

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ**

(повна назва інституту/факультету)

**кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ**

(повна назва кафедри)

**«До захисту допущено»**

Завідувач кафедри БМК

Є.А. Настенко

(підпис)

(ініціали, прізвище)

“ ” \_\_\_\_\_ 2019 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

З напрямку підготовки

**6.050101 «Комп'ютерні науки»**

(код і назва)

на тему: Програмний додаток для підтримки прийняття рішення про  
закупівлю медичного обладнання за сукупністю критеріїв

Виконав: студент **IV** курсу, групи **БС-51**

(шифр групи)

**ТЕРЕЩЕНКО ВОЛОДИМИР ОЛЕГОВИЧ**

(прізвище, ім'я, по батькові)

(підпис)

Керівник *проф. каф. БМК, д.т.н., професор, Файнзільберг Л. С.*

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з розділів ДР

(назва розділу)

( посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Консультант з охорони праці *доц. каф. ОППЦБ, к.т.н. Демчук Г.В.*

(назва розділу)

( посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент

*ас. каф. БМІ, Стасюк Ю.П.*

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2019 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

Інститут (факультет) \_\_\_\_\_ **БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ**  
(повна назва)

Кафедра \_\_\_\_\_ **БІОМЕДИЧНОЇ КІБЕРНЕТИКИ**  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) -

**6.050101 «Комп'ютерні науки» (Інформаційні технології в біології та медицині)**  
(код і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри БМК

\_\_\_\_\_ **Є.А. Настенко**  
(підпис) (ініціали, прізвище)

« \_\_\_\_ » \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**  
**на дипломну роботу студенту**

**ТЕРЕЩЕНКО ВОЛОДИМИР ОЛЕГОВИЧ**

(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ **Програмний додаток для підтримки прийняття рішення про закупівлю медичного обладнання за сукупністю критеріїв**

керівник роботи \_\_\_\_\_ **проф. каф. БМК, д.т.н., професор, Файнзільберг Л. С.**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «27» травня 2019 р. № 1404-С

2. Термін подання студентом роботи \_\_\_\_\_ **12 червня 2019 року**

3. Вихідні дані до роботи \_\_\_\_\_ **База даних, що містить показники 128 пацієнтів прооперованих з приводу вроджених вад серця**

4. Зміст роботи \_\_\_\_\_ **Порівняння існуючих методів вибору найкращої альтернативи, теоретична частина, аналітична частина, розробка програмного продукту, реалізація програмного продукту**

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) \_\_\_\_\_ **рисунки, презентація, таблиці**

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи			
Охорони праці	Демчук Г.В., доц., к.т.н.		

## 7. Дата видачі завдання 20 травня 2019 р.

### Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання на ДР	20 травня 2019р.	
2	Аналіз вітчизняних та зарубіжних літературних джерел	22 травня 2019р.	
3	Розробка теоретичної частини ДР	23 травня 2019р.	
4	Вибір алгоритму прийняття рішень. Розробка програмної реалізації алгоритму прийняття рішень та його оптимізація	24 травня 2019р.	
5	Розробка дизайну програмного продукту	25 травня 2019р.	
6	Створення прототипу програмного забезпечення	26 травня 2019р.	
7	Розробка та тестування програмного забезпечення	27 травня 2019р.	
8	Оформлення практичної частини дипломної роботи	28 травня 2019р.	
9	Розділ ДР з «Безпеки життєдіяльності та охорони здоров'я»	29 травня 2019 р.	
10	Проходження нормоконтролю по оформленню ДР	29 травня 2019р - 7 червня 2019р	
11	Предзахист ДР та допуск до захисту ДР	7-11 червня 2019р	
12	Подання ДР рецензенту. Отримання рецензії.	11-12 червня 2019р	
13	Подання в електронному вигляді ДР та анотації до неї на сайт кафедри.	11-12 червня 2019р	
14	Подання пакету документів по ДР до захисту в ЕК	12-15 червня 2019р.	
15	Захист ДР в ЕК	18-22 червня 2019р	

Студент

(підпис)

В.О. Терещенко

(ініціали, прізвище)

Керівник роботи

(підпис)

Л.С. Файнзільберг

(ініціали, прізвище)

## АНОТАЦІЯ

Метою дипломної роботи була розробка програмного продукту для підтримки прийняття рішень про закупівлю медичного обладнання, тим самим створити спосіб оцінки різних альтернатив за сукупністю критеріїв. Для виконання роботи були використані методи, рішень багатокритеріальних задач, табличний спосіб порівнянь бінарних відношень та функціональні можливості мови програмування C#. В результаті було створено програму, що дозволяє порівнювати альтернатив з багатьма критеріями та оцінювати їх пріоритет.

Структура та обсяг роботи: пояснювальна записка складається із вступу, семи розділів, висновків та списку використаної літератури із 45 джерел. Загальний обсяг дипломної роботи складає: 74 сторінок, основного тексту – 51, ілюстрацій – 30, таблиць – 21.

## **АННОТАЦИЯ**

Целью дипломной работы была разработка программного продукта для поддержки принятия решений о закупке медицинского оборудования, тем самым создать способ оценки различных альтернатив по совокупности критериев. Для выполнения работы были использованы методы, решений многокритериальных задач, табличный способ сравнений бинарных отношений и функциональные возможности языка программирования C#. В результате была создана программа, позволяющая сравнивать альтернатив со многими критериями и оценивать их приоритет.

Структура и объем работы: пояснительная записка состоит из введения, семи глав, заключения и списка использованной литературы из 45 источников. Общий объем дипломной работы составляет 74 страниц основного текста - 51, иллюстраций - 30, таблиц – 21.

## ЗМІСТ

ВСТУП .....	9
СПИСОК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	100
РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ .....	11
1.1. Основні відомості про теорію прийняття рішень .....	11
1.2. Етапи прийняття рішення.....	<b>Ошибка! Закладка не определена.</b>
1.3. Класифікація методів прийняття рішень.....	<b>Ошибка! Закладка не определена.</b>
Висновки до розділу 1 .....	<b>Ошибка! Закладка не определена.</b>
РОЗДІЛ 2. МАТЕРІАЛИ І МЕТОДИ ДОСЛІДЖЕННЯ .....	15
2.1. Багатокритеріальний вибір.....	<b>Ошибка! Закладка не определена.</b>
2.2. Попередня обробка даних .....	<b>Ошибка! Закладка не определена.</b>
2.3. Бінарні відношення .....	<b>Ошибка! Закладка не определена.</b>
2.4. Властивості бінарних відносин .....	<b>Ошибка! Закладка не определена.</b>
2.3. Структурування альтернатив .....	<b>Ошибка! Закладка не определена.</b>
Висновки до розділу 2 .....	26
РОЗДІЛ 3. МЕТОДИ ЗНАХОДЖЕННЯ НАЙКРАЩОЇ АЛЬТЕРНАТИВИ ..	27
3.1. Постановка зачачі.....	27
3.2. Метод контрольної точки .....	27
3.3. Метод поступок .....	29
3.4. Метод головного критерію.....	<b>Ошибка! Закладка не определена.</b>
3.5. Метод ELECTRE .....	31
Висновки до розділу 3 .....	34

РОЗДІЛ 4. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	35
4.1. Розробка вимог до програмного додатку .....	35
4.2. Функціональна модель .....	36
4.3. Діаграма декомпозиції першого рівня .....	37
4.4. Алгоритм роботи .....	<b>Ошибка! Закладка не определена.</b>
4.5. Діаграма варіантів використання ...	<b>Ошибка! Закладка не определена.</b>
4.6. Діаграма станів .....	<b>Ошибка! Закладка не определена.</b>
4.7. Діаграма алгоритму роботи методу ELECTRE	<b>Ошибка! Закладка не определена.</b>
4.8. Діаграма послідовностей .....	<b>Ошибка! Закладка не определена.</b>
4.9. Вибір мови програмування .....	<b>Ошибка! Закладка не определена.</b>
4.10. Складність алгоритму .....	<b>Ошибка! Закладка не определена.</b>
Висновки до розділу 4 .....	45
РОЗДІЛ 5. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	46
5.1. Етапи реалізації .....	46
5.2. Використані типи змінних та конструкції .....	46
5.3. Процес розрахунку індексів в програмі .....	48
5.4. Аналіз даних .....	49
5.5. Windows Forms .....	50
5.6. DataGridView .....	52
5.7. TrackBar .....	53
5.8. Основні елементи ПП .....	55
Висновки до розділу 5 .....	57

РОЗДІЛ 6. ОХОРОНА ПРАЦІ .....	58
Вступ.....	58
6.1. Загальна характеристика приміщення .....	58
6.3. Електробезпека.....	61
6.4. Пожежна безпека.....	62
Висновки до розділу 6 .....	64
РОЗДІЛ 7. ЕКОНОМІЧНА ЧАСТИНА .....	65
1.1. Економічний аналіз варіантів розробки програмного продукту .....	65
1.2. Розрахунок заробітної плати розробника .....	66
Висновки до розділу 7 .....	69
ЗАГАЛЬНІ ВИСНОВКИ .....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71



## ВСТУП

На даний момент в Україні наявна проблема з застарілим медичним обладнанням, у багатьох місцевих лікарнях працює обладнання старше ніж 20-30 років. Якщо у приватних та центральних лікарнях міст ця проблема ще не стоїть так гостро, через достатнє фінансування, то у віддалених лікарнях не вистачає коштів з місцевого для придбання усього необхідного обладнання.

Маючи певні кошти на придбання обладнання потрібно визначити найкращі з можливих варіанти придбання певної апаратури за свої кошти. Маючи базу даних з усіма можливими варіантами придбання, потрібен метод для визначення ліпшого за сумарними характеристиками приладу.

В даній роботі розглядається створення програми для знаходження найкращої альтернативи, серед можливих варіантів.

Актуальність роботи: створення програми для знаходження найкращої альтернативи є актуальною задачею для збільшення ефективності використання коштів і є потрібною не лише для коштовних закупівель, але і для приватного використання.

Задачі, що необхідно вирішити у роботі:

1. Провести аналіз вітчизняних та літературних джерел з поставленої проблематики.
2. Обґрунтувати використання методу ELECTRE як головного методу пошуку кращої альтернативи.
3. Розробити програму для обору найкращої альтернативи на основі методу ELECTRE з зручним інтерфейсом.

## СПИСОК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ТПР – теорія прийняття рішень

ПП – програмний продукт

ОПР – особа що приймає рішення

					<b>БС51.22.1300.1404с.ПЗ</b>	Лист
						10
Змін.	Лист	№ докум.	Підпис	Дата		

# РОЗДІЛ 1

## ОГЛЯД ЛІТЕРАТУРИ

### 1.1. Основні відомості про теорію прийняття рішень

На сьогоднішньому етапі розвитку суспільства, кожна людина постійно приймає низку рішень, кожен день. З розвитком науки та суспільства ціна невірні рішення постійно підвищується і може коштувати великих матеріальних втрат та життів. Неправильний розрахунок дози медичного препарату, не своєчасне лікування, пізно виявлена несправність обладнання – усе ці ситуації та багато інших можуть призвести до летальних наслідків або серйозними травмами для здоров'ям. Також, для багатьох критичних ситуацій важливий ще й час за який потрібно прийняти вірне рішення, бо навіть вірне рішення може не виявитися ефективним, якщо буде прийнято запізно. Для своєчасного правильного прийняття рішення є математична дисципліна «Теорія прийняття рішень», вона забезпечує науково обґрунтований підхід до прийнятті найкращого рішення, серед можливих.

Перед усім слід зауважити, що найкраще рішення залежить перед усім від особи яка його приймає. Різні експертні думки можуть не збігатись і по різному оцінювати пріоритети різних критеріїв вибору.

### 1.2. Етапи прийняття рішення

Після вибору особи яка приймає рішення, потрібно пройти декілька етапів:

- 1.Аналіз проблемної ситуації
- 2.Визначення цілей та пріоритетів
- 3.Отримання необхідної інформації про проблему

4.Формування множини можливих варіантів вирішення проблеми та їх ризику

5.Формування критеріїв оцінки рішень та вибір алгоритму прийняття рішення

6.Порівняння альтернатив за визначеними критеріями

7.Обір найліпшої альтернативи

Для вирішення проблеми потрібно проаналізувати ситуацію та визначити цілі, які потрібно досягнути вибраним рішенням. Відсутність певних відомостей може призвести до неправильного рішення. Важливим є виявлення усіх можливих рішень для конкретної проблемної ситуації. Після цього ОПР, в залежності від поставлених цілей, визначає за якими критеріями буде визначатися, їх важливість та пріоритет альтернатив між собою. Різна експертна думка по-різному оцінює важливість різних обставин проблемної ситуації. Після визначення пріоритетів альтернатив – обирається найкраща.

Важливо розуміти, що теорія прийняття рішень допомагає прийняти найкраще рішення саме в певних умовах, зазвичай невизначених. У житті неможливо мати повні відомості що до проблеми, також не завжди відомі усі можливі варіанти рішення та чіткий та однозначний пріоритет критеріїв, не кажучи про те, що не завжди напевно відомий результат того чи іншого рішення. Тому велика відповідальність лежить саме на ОПР при необхідності їх визначення.

### 1.3. Класифікація методів прийняття рішень

Методи прийняття рішень створені аби обирати найкраще рішення відповідно своєї категорії задач. Задачі поділяються на такі, де кожна альтернатива: має один однозначний результат - визначені умови, має декілька варіантів результату та не має їх вірогідності – невизначені умови, має

декілька варіантів результату та їх вірогідність – в умовах ризику. Також кожен тип задач поділяється на однокритеріальні та багатокритеріальні і для кожного з них є свої методи прийняття рішень.

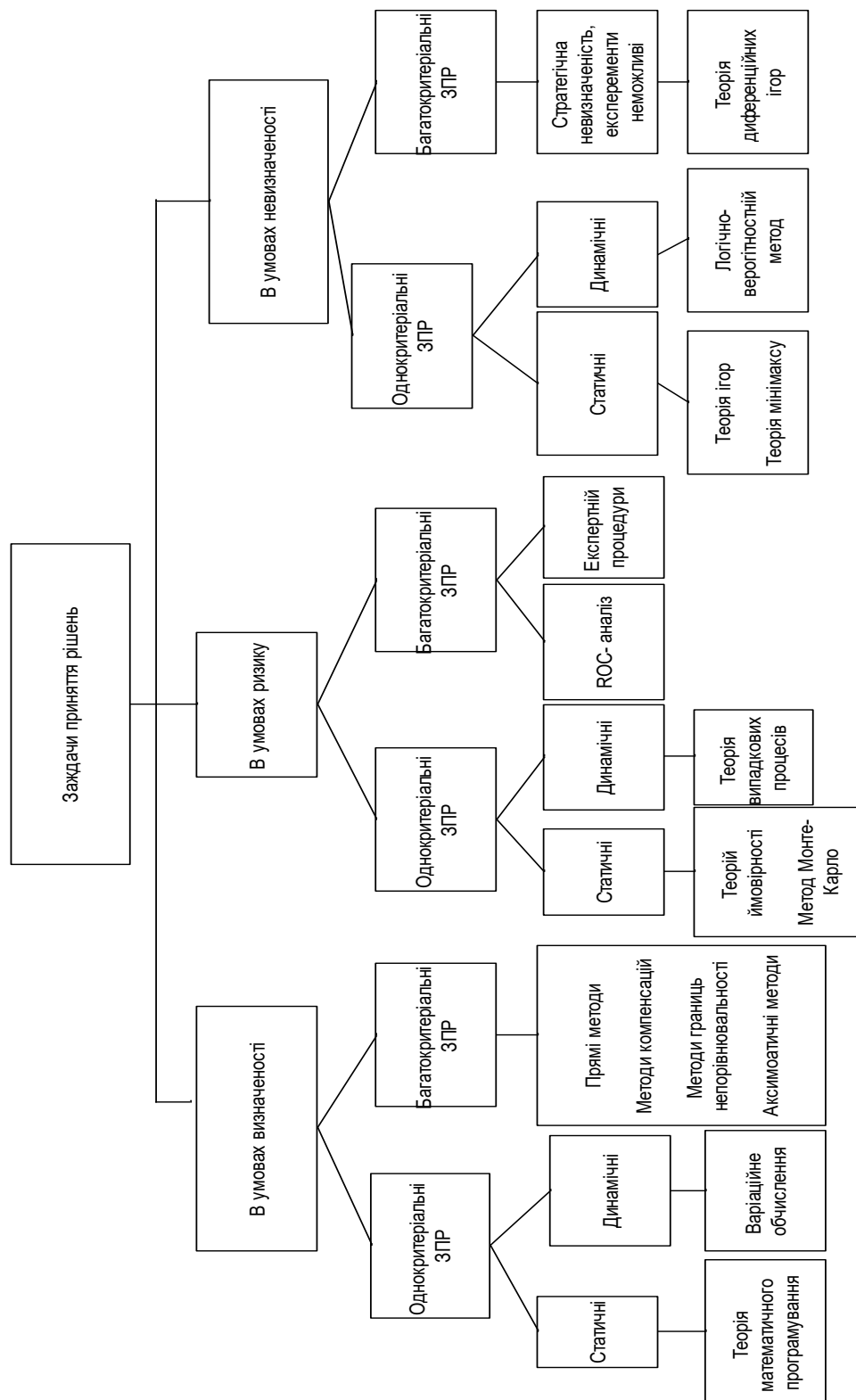


Рисунок 1.1. Класифікація задач з прийняття рішень

Після аналізу задачі слід обрати метод прийняття рішень, залежно від типу задачі, визначити усі можливі варіанти рішення та головні критерії їх оцінок.

Зазвичай кожна ситуація має низку можливих рішень, але не усі серед них мають усі мають братися до уваги при застосуванні методу прийняття рішень. Зазвичай, залежно від умови задачі, серед усіх можливих рішень задачі є так звані аутсайдери - альтернативи, що не є кращими за інші по жодному з критеріїв. В залежності від умов задачі та її типу, альтернативи-аутсайдери можуть бути різними. Відсіюючи такі можна застосовувати метод прийняття рішень.

Різні методи по різному розраховують вплив значення критеріїв на пріоритет альтернативи, тому визначення найліпшого методу для прийняття рішення залежить від ОПР та поставленої цілі.

### **Висновки до розділу 1**

Було отримано поняття про основні поняття з ТПР та основні етапи у алгоритмі прийняття рішень. Також дізнались про класифікацію методів прийняття рішень.

## РОЗДІЛ 2

### МАТЕРІАЛИ І МЕТОДИ ДОСЛІДЖЕННЯ

#### 2.1. Багатокритеріальний вибір

Багатокритеріальні задачі з вибору найліпшої альтернативи – одна з найчастіше виникаюча задача у повсякденному житті. Перс завжди

- Жоден з критеріїв  $q_1(d), q_2(d), \dots, q_p(d)$  не може бути обраний в якості єдиного.
- Одні критерії необхідно максимізувати, а інші – мінімізувати.
- Рішення може виявитися неоптимальним за жодним з критеріїв, але разом з тим, воно має бути найкращим компромісним рішенням з урахуванням всіх критеріїв одночасно.
- Виникає необхідність вибору принципу оптимальності: неоднозначність використання різних принципів оптимальності може призводити до вибору різних альтернатив.
- Виникає неоднозначність впорядкованості критеріїв за важливістю, оскільки ранжування критеріїв залежить від суб'єктивних оцінок ОПР або експертів.
- Виникає необхідність нормування критеріїв, які вимірюються у різних одиницях і діапазонах.
- Виникає питання переходу від якісних критеріїв до кількісних.
- Рішення, оптимальне за одним критерієм, найчастіше не оптимальне за іншими. [1]

Зазвичай неможливо досягти найкращого результату з мінімальними втратами. Досить часто, можна спростити забачу багатокритеріального вибору шляхом переходу від окремих критеріїв  $q_1(d), q_2(d), \dots, q_p(d)$  до суперкритерію  $q_0(d) = q_0[q_1(d), q_2(d), \dots, q_p(d)]$ , шляхом адитивної згортки.

$$q_0(d) = \alpha_1 q_1(d) + \dots + \alpha_p q_p(d).$$

Ми можемо продемонструвати це на прикладі магазину з товарами, який намагається максимізувати свою прибутковість. У магазині є  $p$  різних товарів. Тоді кількість виготовлених товарів рахується як  $q_j(d)$ , де  $j$  – номер товару,  $d$  – фіксоване рішення власника ( $d \in D$ ),  $a_j$  – ціна цього продукту. Лишається лише обчислити суперкритерій та підібрати таке  $d$  для якого, він буде максимальним.

Потрібно пам'ятати, що не завжди можна отримати найліпший результат таким чином, адже продавець у кого в магазині лише оди вид товару, то не можна сподіватись на високі прибутки.

Тому не завжди вірно що можна компенсувати погіршення за одним критерієм збільшенням іншого.

## 2.2. Попередня обробка даних

Для оптимізації роботи з даним табору найкращої альтернативи необхідно зменшити кількість альтернатив, з яких ми обираємо. Для задачі прийняття рішень найкращим способом є застосування множини Парета.

Множина Парета – це підмножина множини альтернатив, в якій кожна альтернатива є кращою за інші альтернативи хоча б за одним з критеріїв(різні характеристики альтернатив - критерій).

Знаходячи множину Парета для нашої множини альтернатив: ми порівнюємо кожен з альтернатив за іншою. Якщо вона одна альтернатива рівна по усім критеріям іншій, а хоча б по одному з них програє, то її можна вилучити з списку альтернатив без втрати для задачі прийняття рішення.



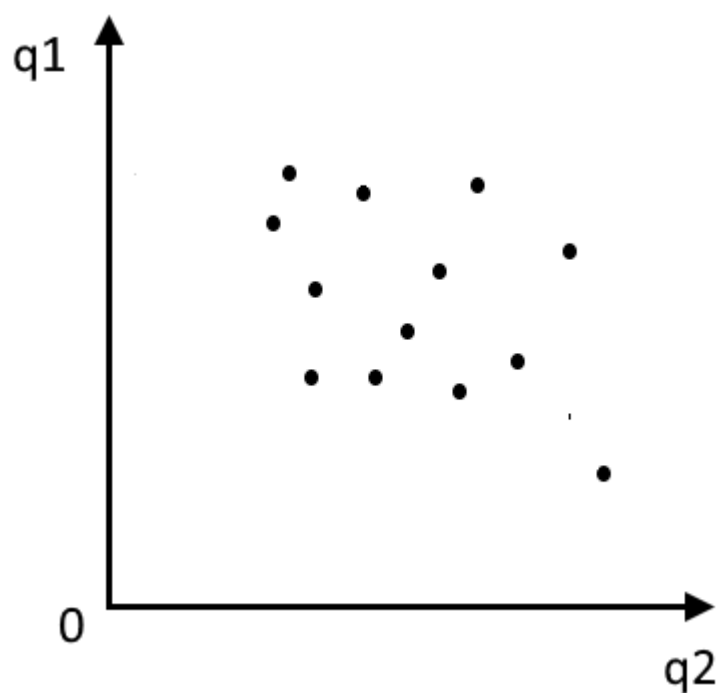


Рисунок 2.1 Зображення альтернатив у системі з двох критеріїв

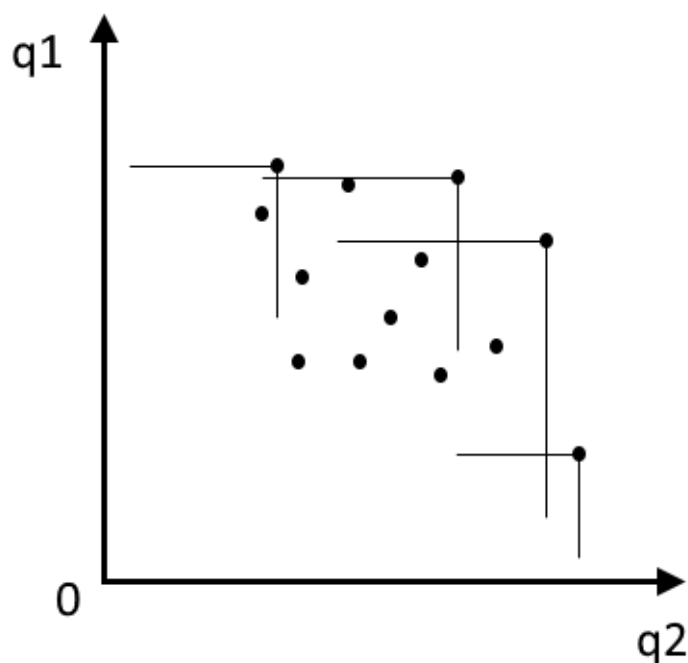


Рисунок 2.2 Графічне зображення того, як визначаються гірші альтернативи(за умови максимізації критеріїв)

Також дуже часто можна вилучити альтернативи в яких значення деяких з критеріїв занадто низьке і альтернатива з таким зазначенням такого критерію не буде підходити при будь яких інших значеннях інших.

Змін.	Лист	№ докум.	Підпис	Дата

**БС51.22.1300.1404с.ПЗ**

Лист

17

### 2.3. Бінарні відношення

Далеко не завжди можна дати характеристику окремій альтернативі  $d \in D$  у вигляді числового критерію або сукупності критеріїв  $q_1(d), \dots, q_p(d)$ , але якщо порівнювати альтернативи одна з одним по парам, то можна визначити яка з них переважає за іншу.

В основі логіки бінарних відношень лежать такі припущення:

1. Окремі альтернативи  $d \in D$  не оцінюють.
2. Для кожної пари альтернатив  $d_i, d_j \in D$ , можна визначити яка з них має перевагу над іншою або що їх неможливо порівняти (вони рівноцінні).
3. Кожне відношення переваги пари альтернатив  $d_i, d_j$  не залежить від інших порівнянь.

Математично бінарне відношення задає на множині альтернатив  $D$  деяку підмножину впорядкованих пар  $(d_i, d_j) \in D \times D$ , для яких виконується певне відношення (relation)  $R$ . Якщо  $d_i, d_j$ , перебувають у відношенні  $R$ , то це записують так:

$d_i R d_j$ , якщо  $d_i, d_j$  знаходяться у відношенні  $R$ ,

$d_i \bar{R} d_j$ , якщо ні.

Визначити бінарне відношення  $R$  означає, що тим чи іншим способом вказані всі пари альтернатив, для котрих виконується  $R$ . Існує три способи визначення бінарних відношень – перелік, граф, таблиця. Найбільш демонстративною з них є граф та таблиця, за якою потім можливо буде зробити ранжування альтернатив.



Рисунок 2.3

Перший спосіб - Перелік: перелічити всі пари  $d_i, d_j$ , для яких виконується певне відношення.

$$d_1 > d_2, d_2 > d_3, d_1 > d_3$$

Другий спосіб - Таблиця: описати бінарні відношення за допомогою таблиці (матриці), елементи якої:

$$r_{ij}(R) = \begin{cases} 1, & \text{якщо } d_i R d_j, \\ 0, & \text{якщо } d_i \bar{R} d_j. \end{cases}$$

та

$$d_i R d_j \Rightarrow d_j \bar{R} d_i.$$

Таблиця 2.1

**Таблиця з прикладом ранжування альтернатив**

Альтернативи	$d_1$	$d_2$	$d_3$
$d_1$	-	1	1
$d_2$	0	-	1
$d_3$	0	0	-

Третій спосіб - Граф: це побудувати графа, в вершинах якого знаходяться альтернатив з множини D. Якщо для вершин  $d_i, d_j$ , виконується відношення і  $d_i R d_j$ , то їх з'єднують дугою, спрямованою від  $d_i$  до  $d_j$ . В іншому випадку дуга відсутня.

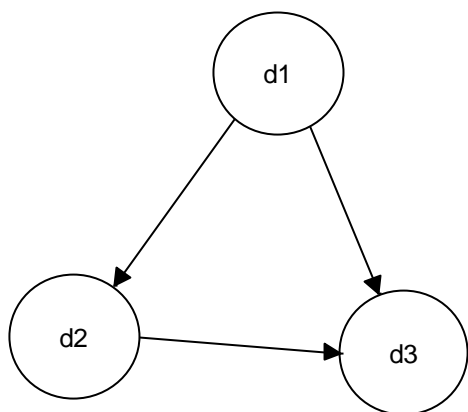


Рисунок 2.4 Побудований граф до таблиці 2.1

Для багатьох альтернатив граф може мати складний вигляд. У такому випадку кращими будуть ті альтернативи, з яких дуги тільки виходять. Зображення переваг у вигляді графів дає змогу наочно оцінювати переваги альтернатив за сукупністю критеріїв.

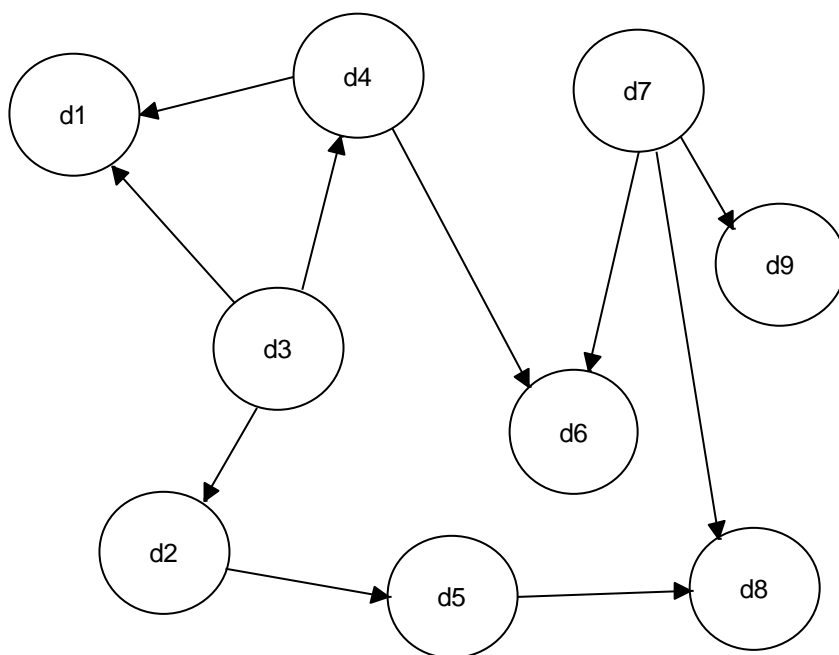
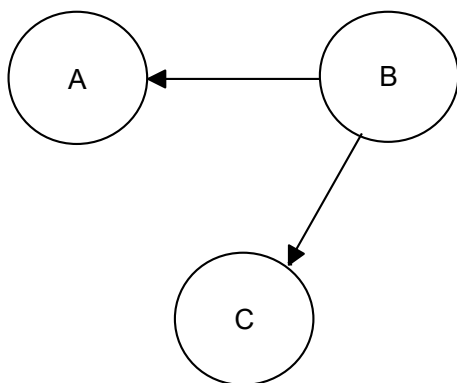
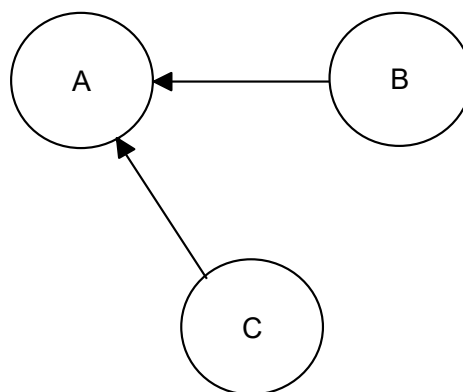
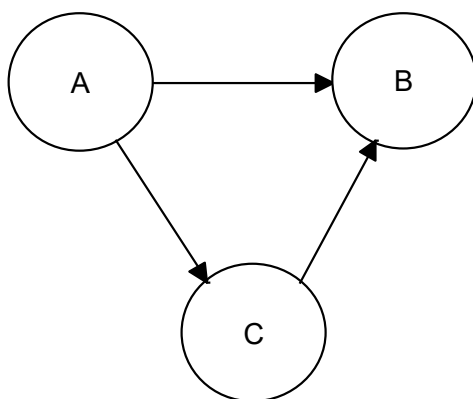


Рисунок 2.5 Приклад складного графу

Приклад порівнянь альтернатив: Нехай три студенти А,В,С здали три екзамени з Фізики, Електротехніки та Програмуванню. Треба порівняти студентів та знайти кращих, за умови, рівноцінності критеріїв.

Таблиця зі значеннями оцінок студентів за іспити

Студент	Дисципліна		
	Фізика	Електротехніка	Програмування
A	3	4	5
B	4	5	3
C	3	5	4

Рисунок 2.6 Граф відношень  
альтернатив за критерієм  
Фізика за таблицею 2.2Рисунок2.7 Граф відношень  
альтернатив за критерієм  
Електротехніка за таблицею 2.2Рисунок 2.8 Граф відношень  
альтернатив за критерієм  
Програмуванню за таблицею 2.2

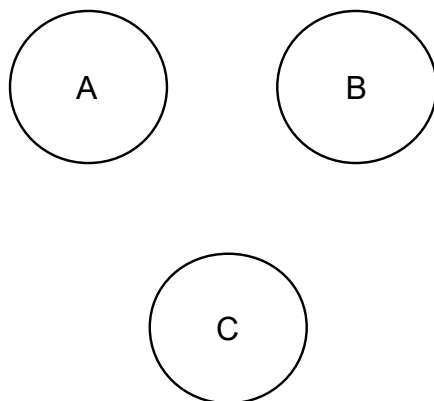


Рисунок 2.9 Загальний граф відношень  
альтернатив таблиці 2.2 за усіма критеріями

Об'єднавши графи та видаливши суперечливі дужки, ми бачимо що жодна з альтернатив не має переваги одна над одною.

Тож тепер змінимо результати з екзамену по Програмуванню

Таблиця 2.2

Таблиця зі зміненими значеннями оцінок студентів за іспити

Студент	Дисципліна		
	Фізики	Електротехніки	Програмуванню
A	3	4	5
B	4	5	5
C	3	5	4

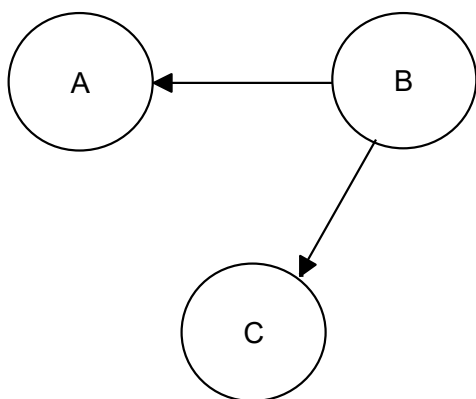


Рисунок 2.10 Граф відношень  
альтернатив за критерієм  
Фізика за таблицею 2.3

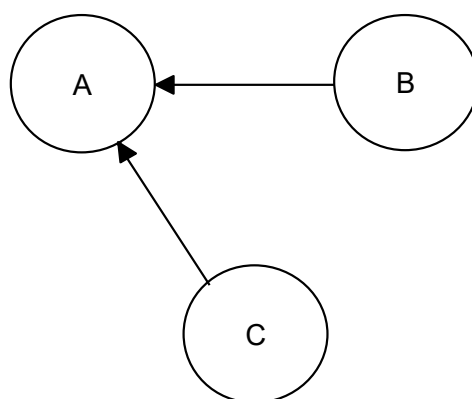


Рисунок 2.11 Граф відношень  
альтернатив за критерієм  
Електротехніка за таблицею 2.3

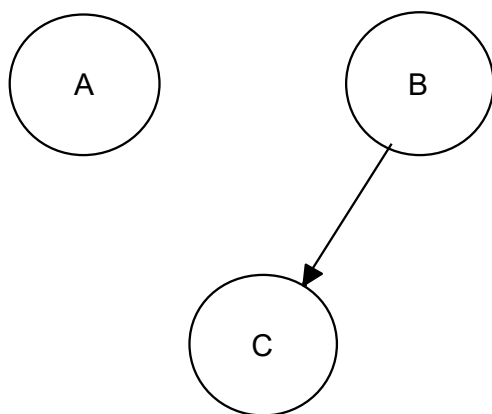


Рисунок 2.12 Граф відношень  
альтернатив за критерієм  
Програмуванню за таблицею 2.3

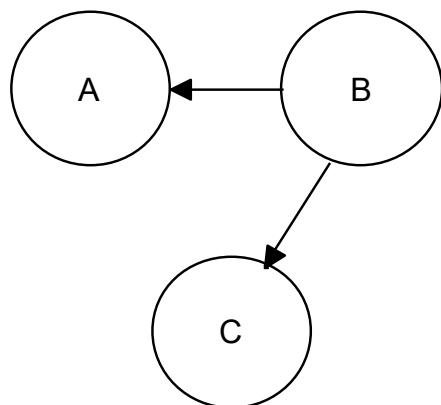


Рисунок 2.9 Загальний граф відношень  
альтернатив таблиці 2.3 за усіма критеріями

Тепер після об'єднання графі, студент В має перевагу над іншими. А знання студентів А та С між собою, можна назвати рівними.

## 2.4. Властивості бінарних відносин

Властивості бінарних відношень  $R$  для елементів  $d_i, d_j \in D$ .

- Бінарне відношення рефлексивне, якщо

Змін.	Лист	№ докум.	Підпис	Дата

**БС51.22.1300.1404с.ПЗ**

Лист

23

$$d_i R d_i,$$

елемент сам з собою знаходиться у відношенні R.

- Бінарне відношення R антирефлексивне, якщо

$$d_i \bar{R} d_i,$$

тобто відношення R є вірним тільки для елементів, що не співпадають.

- Бінарне відношення R симетричне, якщо

$$d_i R d_j \Rightarrow d_j R d_i, \forall d_i, d_j \in D,$$

тобто з того, що елемент  $d_i$  знаходиться у відношенні R з елементом  $d_j$  випливає, що елемент  $d_j$  знаходиться у відношенні R з елементом  $d_i$ .

- Бінарне відношення R асиметричне, якщо

$$(d_i R d_j) \wedge (d_j R d_i) \Rightarrow d_i = d_j,$$

тобто з того, що елемент i d знаходиться у відношенні R з елементом j d випливає, що елемент j d не знаходиться у відношенні R з елементом i d

- Бінарне відношення R антисиметричне, якщо

$$(d_i R d_j) \wedge (d_j R d_i) \Rightarrow d_i = d_j,$$

тобто з того, що елемент  $d_i$  знаходиться у відношенні R з елементом  $d_j$ , а елемент  $d_j$  знаходиться у відношенні R з елементом  $d_i$  випливає, що  $d_i$  та  $d_j$ .

- Бінарне відношення R транзитивне, якщо

$$(d_i R d_j) \wedge (d_j R d_z) \Rightarrow d_i R d_z, \forall d_i, d_j, d_z \in D,$$

тобто з того, що елемент  $d_i$  знаходиться у відношенні R з елементом  $d_j$ , а елемент  $d_j$  знаходиться у відношенні R з елементом  $d_z$  випливає, що елемент  $d_i$  знаходиться у тому ж відношенні R з елементом  $d_z$ .

Таким чином  $d_i$  та  $d_j$  є еквівалентні, якщо R одночасно є рефлексивною, симетричною та транзитивною.



## 2.5. Структурування альтернатив

Завдяки бінарним властивостям пар альтернатив  $d_i$ ,  $d_j \in D$ , можна визначити пріоритетність альтернатив між собою (ранжувати їх).

Найпростіший метод ранжування – це метод рядкових сум, для нього потрібно побудувати таблицю бінарних відношень альтернатив. Таблицю слід заповнювати таким чином:

- Якщо альтернатива у рядку краще, за альтернативу у стовпчику, то слід писати 1 у відповідній клітинці
- Якщо альтернатива у рядку гірша, за альтернативу у стовпчику, то слід писати 0
- Якщо альтернативи неможливо порівняти або вони еквівалентні то слід писати 0,5

Після заповнення таблиці слід просумувати значення кожного рядку. Результат – це вага кожної альтернативи, найкраща альтернатива має найбільшу вагу та 1 ранг, а усі інші – відносно їх вазі.

Таблиця 2.3

**Таблиця зі значенням бінарних відношень**

	$d_1$	$d_2$	$d_3$	$d_4$	$\Sigma$
$d_1$	-	0	0,5	1	1,5
$d_2$	1	-	0,5	0	1,5
$d_3$	0,5	0,5	-	0	1
$d_4$	0	1	1	-	2

У таблиці 2.3 показане бінарне відношення альтернатив за яким альтернатива  $d_4$  має найвищу вагу, отож має 1 ранг, далі альтернативи  $d_1, d_2$  мають рівні значення ваг, тож їх ранг – 2, а альтернатива  $d_3$  відповідно 3.

## Висновки до розділу 2

Для вирішення поставлених задач були використані теоретичні відомості стосовно бінарних відносин та методами попередньої обробки даних. У реалізації ПП були дотримані усі визначення багатокритеріальних задач з вибору найкращої альтернативи. Розглянуті особливості методів їхнього рішення.

					<b>БС51.22.1300.1404с.ПЗ</b>	Лист
						26
Змін.	Лист	№ докум.	Підпис	Дата		

## РОЗДІЛ 3

### МЕТОДИ ПОШУКУ НАЙКРАЩОЇ АЛЬТЕРНАТИВИ

#### 3.1. Постановка зачачі

Постановка задачі з вибору найкращої альтернативи передбачають обробку альтернатив з декількома характеристиками, що мають власну вагу. Жодну з характеристик не можна прийняти за єдиний за єдиний критерій, отже тоді ми маємо критеріїв  $q_1(d), \dots, q_p(d)$ ,  $p$  – кількість критеріїв. Це значить що ми маємо задачу багатокритеріального вибору. Для рішення таких задач ми розглядаємо множину альтернатив  $D\{d\}$ , яку ми зводим до підмножини Парето. Було розглянуто декілька підходів для розв'язання такого роду задач:

#### 3.2. Метод контрольної точки

При оборі найкращої альтернативи ми можемо оцінювати їх не по вазі різних критеріїв відносно один одного, а відносно характеристик еталонного значення – контрольної точки. Задача зводиться до того, аби характеристики альтернативи були як умово ближчі до значень контрольної точки. Відстань від альтернативи.

На рисунку зображена контрольна точка(КТ) та чотири альтернативи  $d_1, d_2, d_3, d_4$ , що зображені на площині двох критеріїв  $q_1$  та  $q_2$ , значення яких слід максимізувати. Метод контрольної точки визначає найкращу альтернативу, вираховуючи «відстань» між контрольною точкою та альтернативою. Альтернатива що має найменшу відстань до контрольної точки є найліпшою. З рисунку 3.1 очевидним є, що найближчою альтернативою, за значенням критеріїв, є альтернатива  $d_3$ . Проте навіть без використання метрик можна побачити, що альтернатива  $d_1$  та  $d_2$  кращі за неї.

Вони мають більше значення критеріїв і є ліпшими альтернативами, тому використання такого методу не завжди є оптимальним.

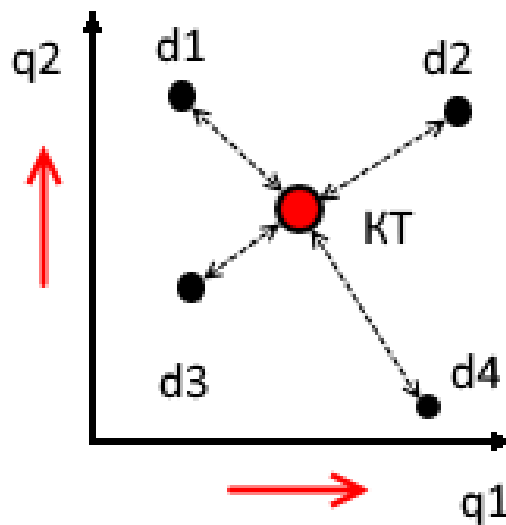


Рисунок 3.1 Зображення чотирьох альтернатив та контрольної точки у системі з двох критеріїв  $q_1$  та  $q_2$

Відстань між альтернативами можна вимірювати різними способами. Для двох альтернатив  $p$  та  $q$ , з критеріями  $(p_1, p_2, \dots, p_n)$  та  $(q_1, q_2, \dots, q_n)$  відповідно. Зазвичай використовують евклідову відстань, що відповідає природній властивості відстані та обраховується за теоремою Піфагора:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Також часто використовується «манхетонська» відстань, вона використовується тоді, коли відстань між альтернативами дорівнює суммі відстаней по кожному з критеріїв. Найбільш наочно це показує маршрут автомобіля, який їздить до кінцевої точки найкоротшим шляхом.

$$d(p, q) = \sqrt{\sum_{i=1}^n |p_i - q_i|}$$

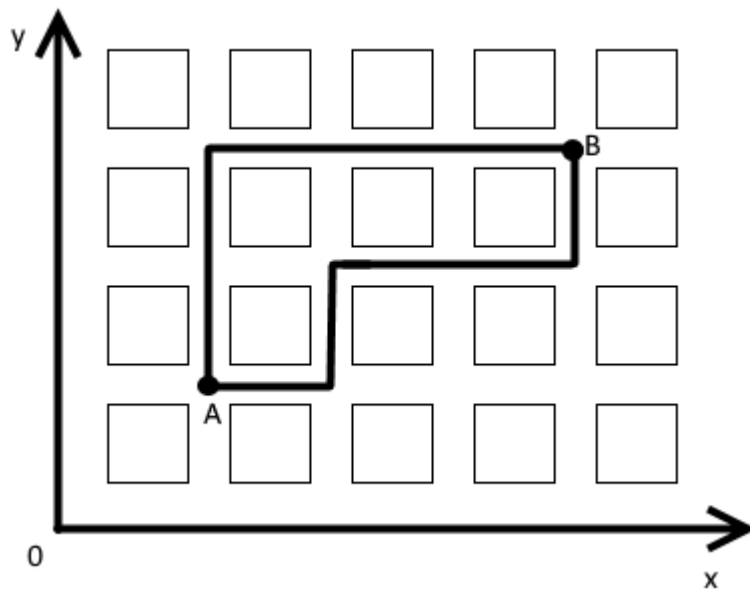


Рисунок 3.2 рисунок можливих відстаней від точки А до точки В за манхетонською метрикою

### 3.3. Метод поступок

Метод поступок передбачає, що всі критерії можна впорядкувати за важливістю для ОПР. Це значить що уся критерії  $q_1(d), \dots, q_p(d)$  можна впорядкувати таким чином.

$$q_1(d) > q_2(d) > \dots > q_p(d)$$

Тепер ми знаходимо множину найліпше значення критерію  $q_1(d)$  робимо поступку  $\Delta_{q1}$ . Потім шукаємо найліпше значення для  $q_2(d)$  серед альтернати для яких  $q_1(d) \geq q_1(\max) - \Delta_{q1}$ . Для знайденого значення знов робимо поступку  $\Delta_{q2}$ . Таким чином продовжуємо до останнього критерію (для нього поступок не робимо) і залишається одна альтернатива.

Сутність методу полягає в тому, що виграш за менш важливим критерієм повинен перевищувати програш за більш важливим. Метод є ефективним тоді, коли роблячи деяку поступку ми отримуємо виграш наступного по важливості критерію на більшу величину.

Змін.	Лист	№ докум.	Підпис	Дата

БС51.22.1300.1404с.ПЗ

Лист

29

Даний підхід послідовно відкидає альтернативи, що не підходять під надані обмеження, і є досить ефективним при необхідності обору підходящих альтернатив. Але він не може бути зручно зображеним у користувацькому інтерфейсі, через невизначеності кількості критеріїв, а також через необхідність мати професійні знання стосовно предмету обору(треба мати більше глибокі знання стосовно об'єктів які ти вибираєш, щоб знати яку поступку можна робити). Тому даний метод не підходить для вирішення задачі і може слугувати лише як допоміжний.

### 3.4. Метод головного критерію

Один з критеріїв вибирається головним  $q_1(d) > q_2(d), \dots, q_p(d)$ , а потім на усі інші критерії вводиться обмеження. Після знаходження підмножини альтернатив, що задовольняє обмеженням, ми вирішуємо однокритеріальну задачу, де шукаємо альтернативу за максимальним значенням  $q_1(d)$ .

Для прикладу зробимо таблицю з різними товарами, припустимо на потрібно купити певний товар як умово дешевше, та найбільше, але також він повинен мати великий термін придатності.

Таблиця 3.1

**Таблиця з даних зі значенням критеріїв для демонстрації методу головного критерію**

Варіанти	Критерії		
	Ціна(грн)	Вага(кг)	Термін придатності(міс)
1	30	2	12
2	20	5	8
3	50	3	12
4	30	7	10
5	70	11	10
7	60	9	6
8	65	8	15

Зробимо наступні обмеження для значень критеріїв за таблиці 3.1

- Ціна не більше 50 грн.
- Вага не менше 3 кг.
- Термін придатності не менше за 10 місяців

Таким чином в нас залишилися лише 3 та 4 альтернативи , серед яких ми обираємо 4, так як в неї найменша ціна.

Таблиця 3.2

**Таблиця з обмеженими даними зі значенням критеріїв для  
альтернатив таблиці 3.1**

Варіанти	Критерії		
	Ціна(грн)	Вага(кг)	Термін придатності(міс)
3	50	3	12
4	30	7	10

Головна проблема цього методу, що після вводу обмежень, в нас може не залишитися жодної альтернативи. Також він є не об'єктивним, якщо припустити, що найкраща альтернатива визначається матиме більший виграш при високих значеннях інших критеріїв. Медичне обладнання не може бути визначено лише однією характеристикою - отже цей метод не підходить для вирішення поставленої задачі.

### 3.5. Метод ELECTRE

Метод ELECTRE – це евристичні методи ранжування альтернатив. В основі цього методу лежить принцип, що жоден з критеріїв не може бути головним, цей підхід не дає оптимальне рішення за жодним з критеріїв, але воно буде найкращим компромісним рішенням, з урахуванням всіх критеріїв. Особливістю евристичних методів є залежність від особи, що приймає рішення, яка обирає важливість критеріїв.

Метод розв'язує багатокритеріальну задачу шляхом створення двох індексів(згоди та незгоди) для обору кращого варіанту за ними, з урахуванням думки особи що приймає рішення. Спочатку кожному з критеріїв  $q_1, \dots, q_p$  ставиться у відповідність вага  $\omega_\mu, \mu = 1, \dots, p$ , що визначає його відносну важливість відносно ОПР, вона вимірюється цілим додатнім числом, та може бути однаковою у різних критеріїв(тоді вони однаково важливі).

Таблиця 3.3

### Приклад даних для методу ELECTRE

Альтернатива	Вага критеріїв		
	$\omega_1=5$	$\omega_2=3$	$\omega_3=2$
A	5	3	4
B	5	4	3
C	4	5	3

Потім для кожної пари альтернатив  $d_i, d_j \in D$ , множину всіх критеріїв  $I=\{q_1, \dots, q_p\}$  розбивають на три підмножини  $I^+, I^=, I^-$ , що не перетинаються:

- $I^+$  - підмножина критеріїв, за якими  $d_i > d_j$
- $I^=$  - підмножина критеріїв, за якими  $d_i = d_j$
- $I^-$  - підмножина критеріїв, за якими  $d_i < d_j$

Далі йде розрахунок індексів згоди  $\alpha_{ij}$  та незгоди  $\beta_{ij}$  на основі парного порівняння альтернатив.

Індекс згоди  $\alpha_{ij}$  показую наскільки ця альтернатива ліпша іншу  $d_i > d_j$ , рахується шляхом відношення суми ваг критеріїв  $q_1, \dots, q_p$  критеріїв з підмножин  $I^+$  та  $I^=$  до загальної суми ваг, (отримується число від 0 до 1).

$$\alpha_{ij} = \frac{\sum_{\mu \in I^+ \cup I^=} \omega_\mu}{\sum_{\mu=1}^p \omega_\mu}$$

$$1 \leq \alpha_{ij} \leq 0$$

З виразу випливає, що  $\alpha_{ij} = 1$  якщо  $I^- = \{ \}$ , тобто відсутній критерій за яким  $d_i < d_j$ .



Таблиця індекси згоди

	A	B	C
A	-	0,7	0,7
B	0,8	-	0,7
C	0,3	0,5	-

Індекс незгоди  $\beta_{ij}$  показує наскільки сильно ця альтернатива гірша за іншу  $d_i < d_j$  по одному з критеріїв, він визначається на основі самого суперечливого критерію за яким  $d_i < d_j$  найбільшою мірою. Спочатку кожна з альтернатив порівнюється з кожною іншою, і рахується значення різниці критерію за яким перша альтернатива сильніше за все програє другій, потім результат ділиться на діапазон значення критерію  $\Delta_\mu$ , це забезпечує нормування значень.

$$\beta_{ij} = \max_{\mu \in I^-} \left| \frac{z_\mu(d_l) - z_\mu(d_i)}{\Delta_\mu} \right|$$

$$1 \leq \beta_{ij} \leq 0$$

$z_\mu(d_l), z_\mu(d_i)$  – числова оцінка альтернатив  $d_l, d_i$  за  $\mu$  критерієм.

Таблиця 3.5

Таблиця індексів незгоди

	A	B	C
A	-	0,5	1
B	0,5	-	0,5
C	0,5	0,5	-

Останній етап – ОНР обирає яке саме мінімальне значення рівня згоди  $\alpha^0$  і максимальне значення рівня незгоди  $\beta^0$  може приймати альтернатива аби можна було сказати, що вона краща за іншу. Альтернатива  $d_i > d_j$ , якщо  $\alpha_{ij} \geq \alpha^0, \beta_{ij} \leq \beta^0$ . На основі двох таблиць будується граф і відкидаються гірші альтернативи та залишаються лише ті, що не можна порівняти.

Таким чином при  $\alpha^0 = 0,8$  і  $\beta^0 = 0,5$

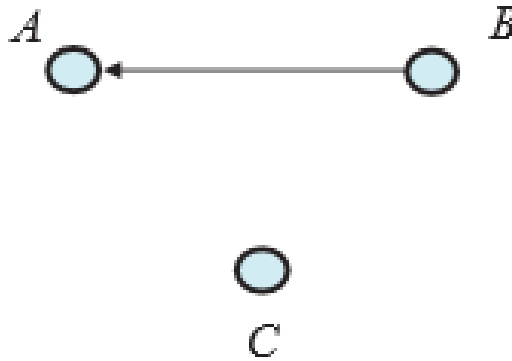


Рисунок 3.3. Граф відношень альтернатив за таблиці 3.3

Великим недоліком цього методу є не можливість обирати не підготовлені дані та опрацьовувати їх великий обсяг, через велику обрахункову складність. Спочатку потрібно звужити число альтернатив, аби обов'язково застосувати множину Парета (одна альтернатива краща або така сама, як друга по усім критеріям), та при можливості обмежити множину альтернатив, я було наведено у методі головного критерію.

Перевага методу ELECTRE – можливість індивідуально обрати найкращу альтернативу серед наведених та варіативність для ОПР. Тому цей метод було використано при створення програми.

### Висновки до розділу 3

Було порівняно різні методи порівняння альтернатив, визначені їх переваги та недоліки. Був обраний метод ELECTRE, як найліпший метод обору найліпшої альтернативи та оптимальний для програмної реалізації.

## РОЗДІЛ 4

### ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 4.1. Розробка вимог до програмного додатку

Для початку реалізації програмного додатку потрібно спроектувати його структуру. Проаналізувати які данні будуть потрапляти на вхід програми, що буде отримано у результаті та які ресурси для цього потрібні.

Під час аналізу функціональних можливостей, що повинна мати програма, був створений типовий функціонал, який підтримує програмний продукт:

1. Можливість роботи з різними файлами, їх обробка та збереження.
2. Розрахунок коефіцієнтів згоди та незгоди, на основі наданих баз даних.
3. Можливість оптимізації та визначення вагомості критеріїв для подальшої обробки програмою.
4. Візуальне подання даних відносної пріоритетності альтернатив.
5. Регулювання граничних показників згоди та незгоди для фільтрованого виводу оптимальних альтернатив.

Оскільки ціль програми – поміч у прийнятті рішень, то цільова аудиторія – люди відповідальні з закупівлі медичного обладнання та апаратури. Також користувачу не потрібно володіти спеціальними навиками для використання програми.

## 4.2. Функціональна модель

Для створення функціональної моделі, що відображає структуру і функції системи, зв допомогою AllFusion Process Modeler, була побудована контекстна діаграма.

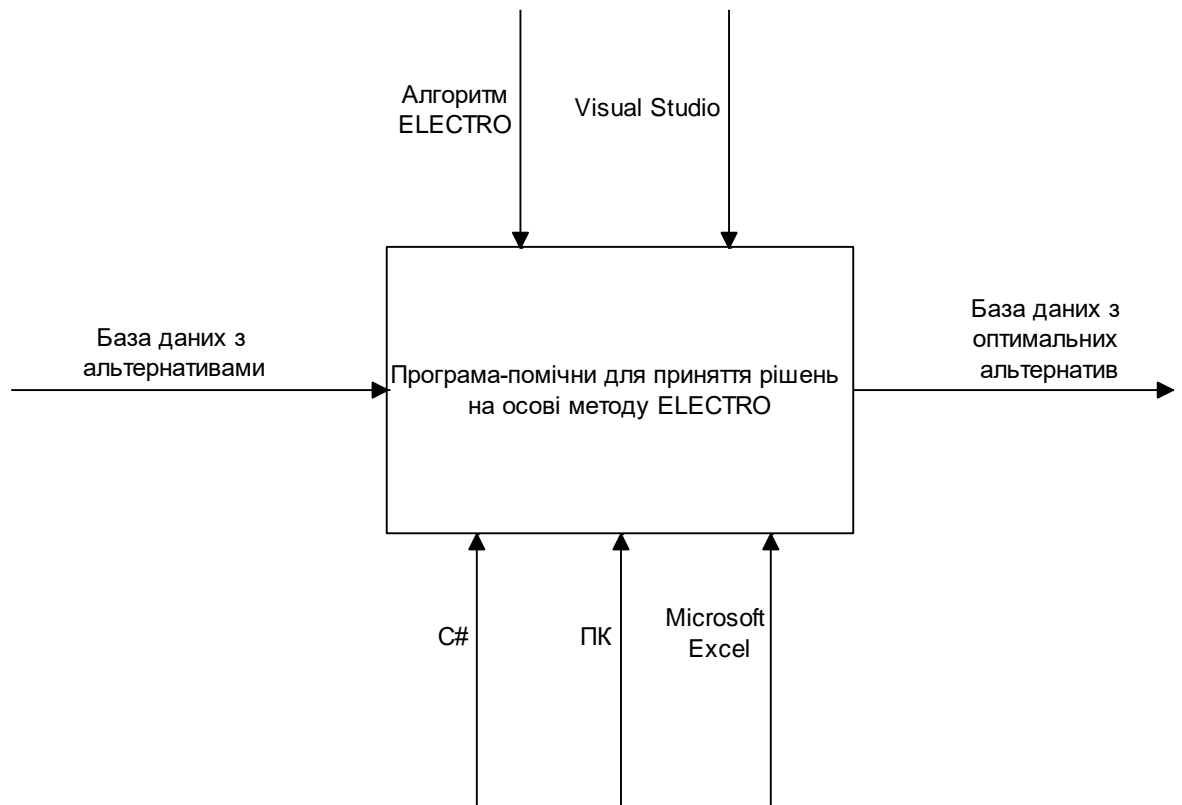


Рисунок 4.1. Контекстна діаграма IDEF0

На діаграмі видно, що на вхід ПП потрапляє база даних за альтернативами, а на виході ми отримуємо вже базу з обчисленими даними де вказані найкращі альтернативи. Зверху відображені методи завдяки яким був реалізований ПП, з низу вказані ресурси які необхідні програмі для функціонування.

### 4.3. Діаграма декомпозиції першого рівня

Дана діаграма відображає роботу ПП розбиту на під процеси . На вхід ПП потрапляє база даних з альтернативами, їх форматують у той вид, у якому програма може буде працювати з ними, також на цьому етапі відсіюються пошкоджені або непридатні для обробки дані. Далі користувач налаштовує параметри для обробки даних відносно поставлених цілей та відправляє зменшену(або таку саму, при не внесенні зміні) базу даних на обробку методу, та отримує базу найліпших рішень. Вихідні та відформатовані дані можна зберегти у файл.

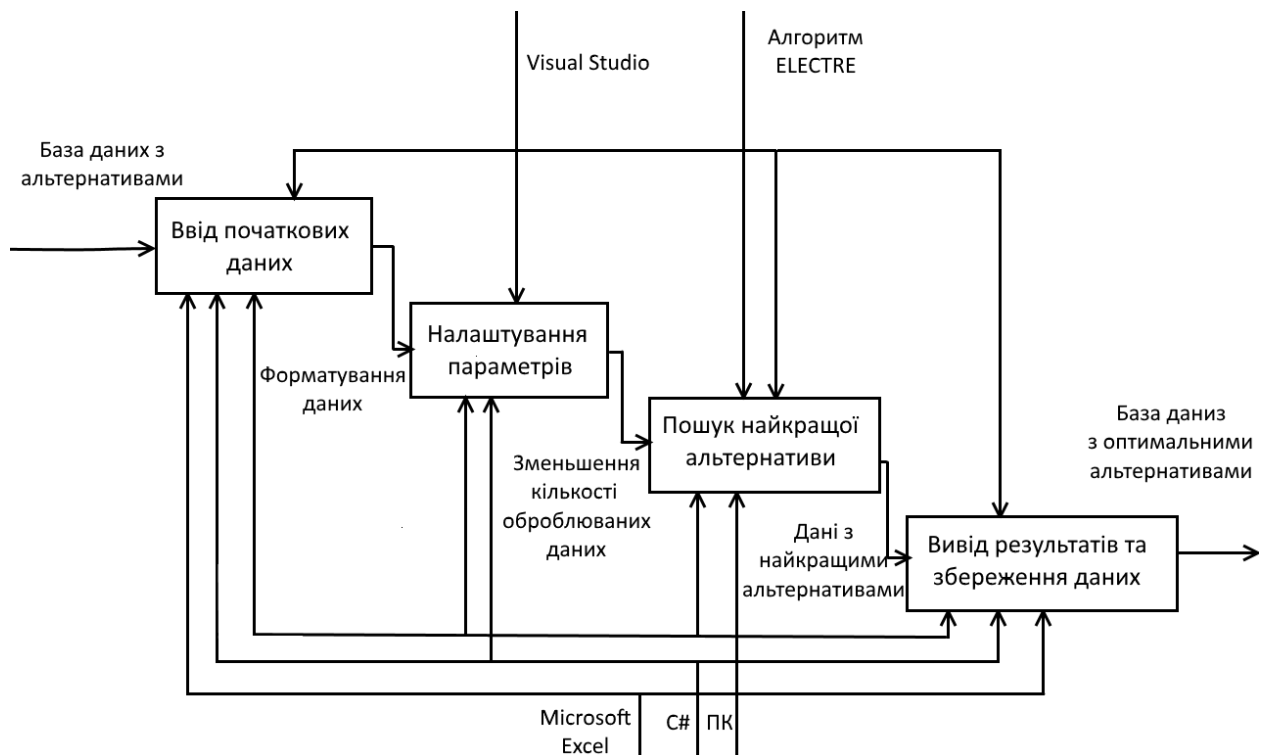


Рисунок 4.2. Діаграма декомпозиції першого рівня

В якості методів пошуку найкращої альтернативи був використаний метод прийняття рішень ELECTRO. Виходом є таблиця бінарних відношень за вагами альтернатив.

#### 4.4. Алгоритм роботи

Наступна діаграма показує алгоритм роботи ПП розбитий на дії, для реалізації методу з пошуку найкращої альтернативи.



Рисунок 4.3. Алгоритм роботи ПП

#### 4.5. Діаграма варіантів використання

На цій діаграмі показано можливі варіанти дій користувача під час використання програми у певний момент часу. Усі можливості ПП доступні у що най більше у два кроки. В купі з зрозумілим меню, це робить процес використання ПП досить зручним та зрозумілим для звичайного користвача.



Рисунок 4.4. Діаграма варіантів використання

#### 4.6. Діаграма станів

На наступній діаграмі показані стани програми у яких вона перебуває під час роботи. З діаграми видно, що найчастіше ПП знаходиться у стані очікування дій користувача, переважно вводу даних та налаштувань.

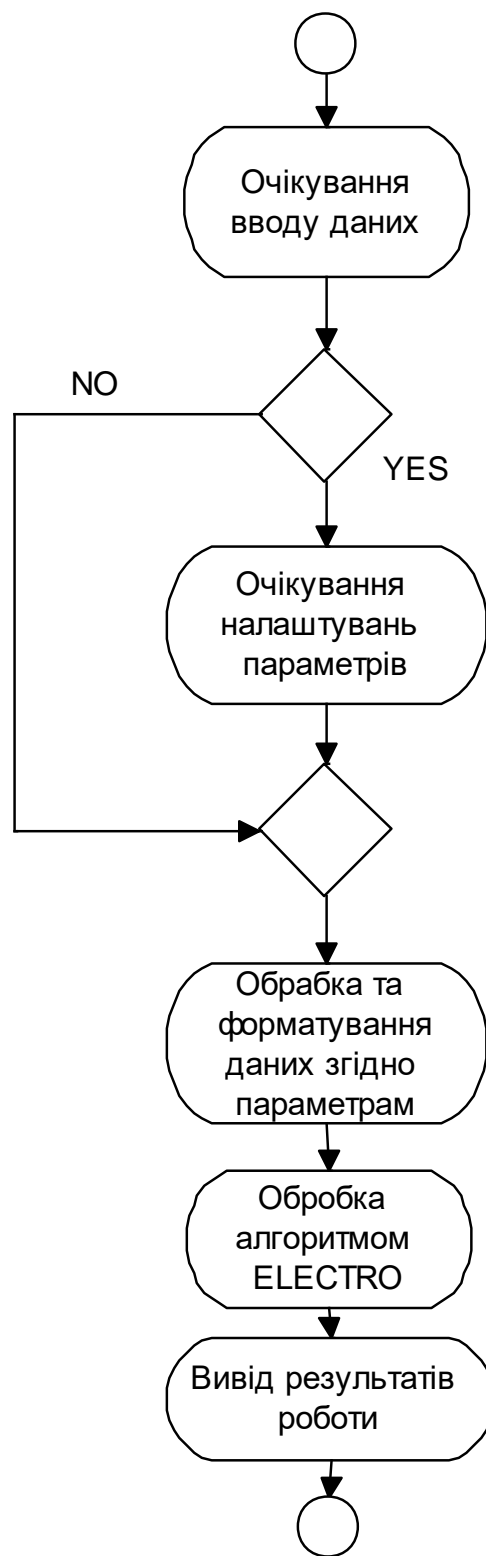


Рисунок 4.5. Діаграма станів



#### 4.7. Діаграма алгоритму роботи методу ELECTRE

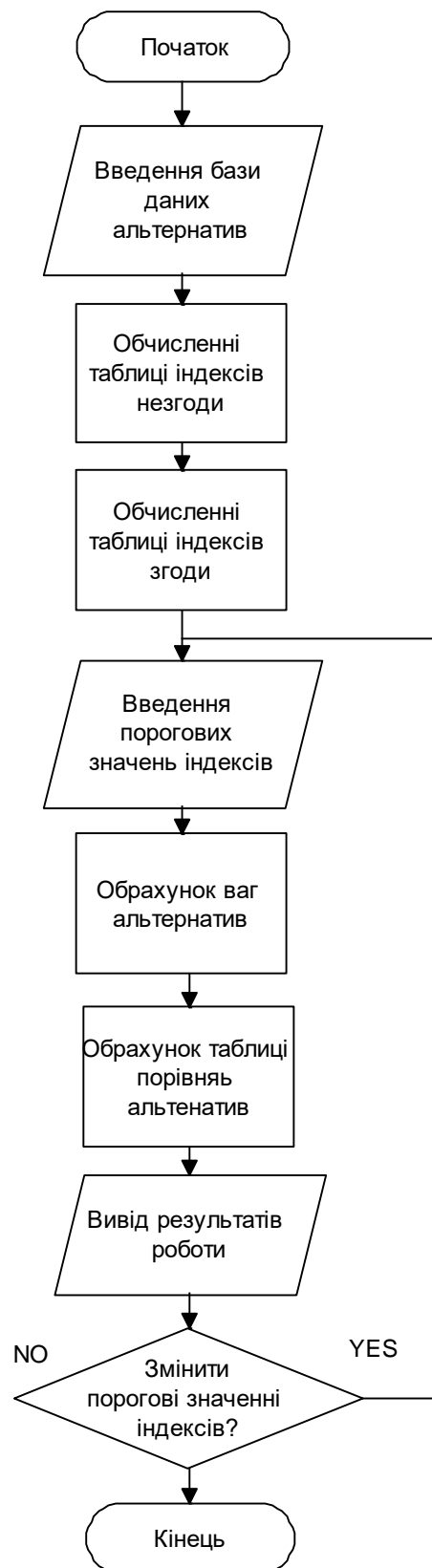


Рисунок 4.6. Алгоритм роботи методу ELECTRE

Дана діаграма показує послідовність дій ПП при обчисленні результатів, для методу ELECTRE. Видно, що алгоритм розраховує перші етапи без участі користувача та розраховує таблиці використовуючи лише базу даних, що потрапила на вхід. Далі на діаграмі зображена особливість роботи методу, завдяки якому він є досить гнучким. Залежно від порогових значень індексів згоди та незгоди, які користувач може змінювати, різні альтернативи по різному взаємодіють один з одним у таблиці порівнянь альтернатив. Вони можуть переважали одна одну лише якщо їхні значення індексів згоди та незгоди підходять введеним пороговим значенням.

На основі результатів таблиці порівнянь альтернатив, розраховуються ваги альтернатив. Користувач може змінювати порогові значення, таким чином поки його не задовільнить результат обробки даних.

#### 4.8. Діаграма послідовностей

Діаграма послідовностей показує послідовність взаємодій елементів ПП за часом. Дана діаграма доволі проста, оскільки усі етапи роботи програми йдуть послідовно отожд і елементе теж будуть так задіяні. Важило зауважити, що ввід параметрів для роботи ПП не є обов'язковим, тому як і запис результатів роботи у файл.

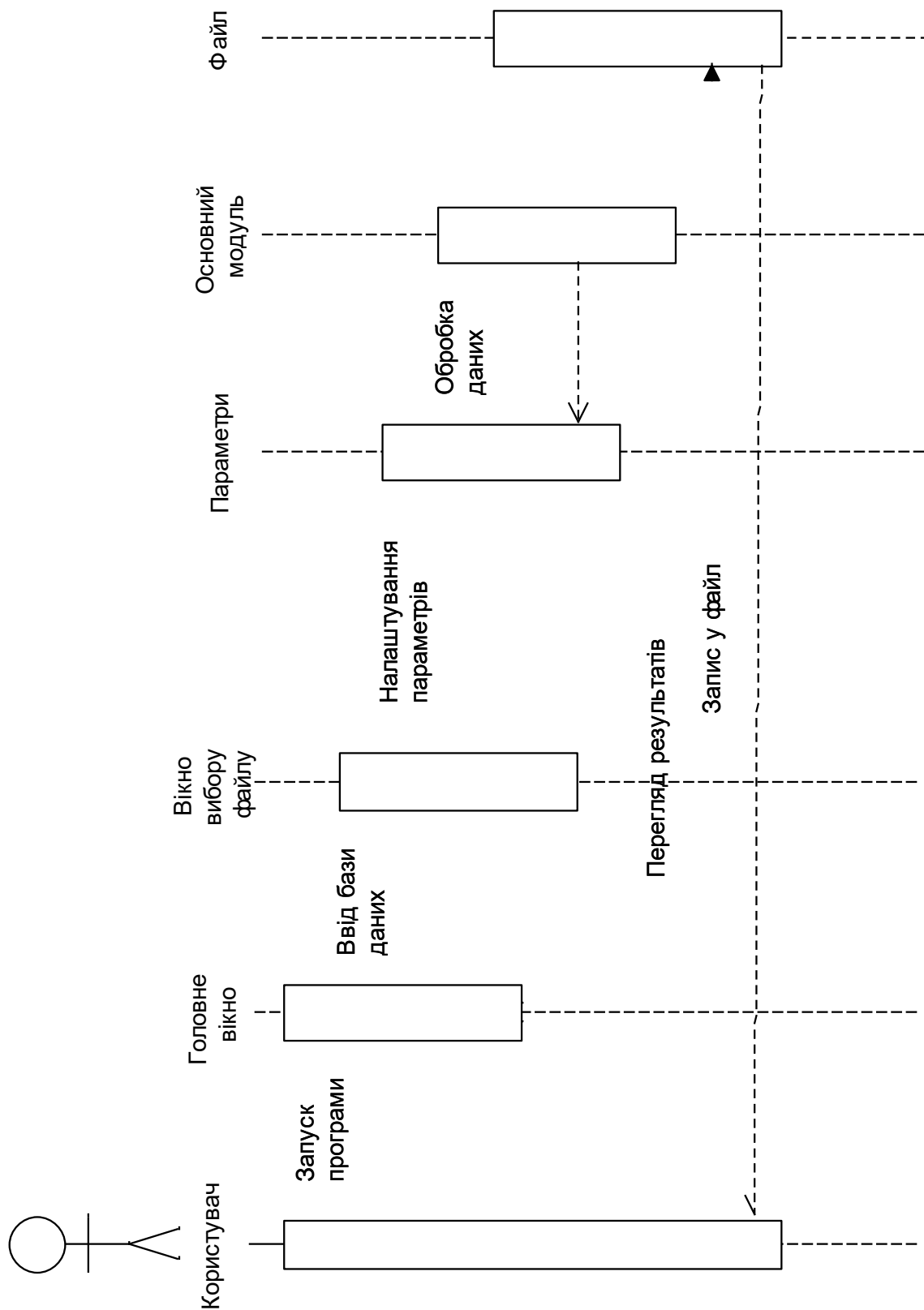


Рисунок 4.7. Діаграма послідовностей

#### 4.9. Вибір мови програмування

Мова програмування - це штучна мова, створена для передачі команд машинам, зокрема комп'ютерам. Їх використовуються для створення програм, бо вона є інструкцією, послідовністю дій, які повинен виконати пристрій.

Більш строге визначення: мова програмування — це система позначень для опису алгоритмів та структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір синтаксичних та лексичних правил, різні правила розрізняють зовнішній вид коду тапливають на її функціональні можливості, переваги та недоліки.

C# - об'єктно-орієнтована мова програмування, створена у 2001 році розроблена корпорацією Microsoft. Це мова з C-подібним синтаксисом, що має статичну типізацію(тип змінних повинен бути визначений відразу під час їх створення) та підтримує поліморфізм, перевантаження операторів, атрибути, події, властивості, делегати, методи та узагальнені типи.

Через те, що ПП повинен буде опрацьовувати великі розрахунки, для його ефективної роботи потрібна була швидка обробка даних. Тому було вирішено обирати серед мов, що компілюють код перед запуском програми. Також перевагою C# є наявність спеціалізованої середи розробки Microsoft Visual Studio та спрямованість ПП на ОС Windows. Серед розробки має необхідні вистроєні бібліотеки та уловлювач помилок, що прискорило розробку ПП.

#### 4.10. Складність алгоритму

Реалізувавши метод ELECTRE його приблизна обчислювальна складність дорівнює  $2N^2 T$  де N-кількість альтернатив, T-кількість критеріїв.

Це доволі велика обчислювальна складність(порівняно з методом поступок, що має майже лінійну складність що залежить від кількості критеріїв). Через це були додані методи оптимізації даних, такі як множена Парета, та попередній ввід обмежень.

Також для більшого звуження множини альтернатив у ПП була додана можливість вилучати з урахування окремі критерії, та вводити на них обмеження.

#### **Висновки до розділу 4**

В даному розділі було спроектований програмне забезпечення ПП, а саме:

- Була побудована діаграма роботи алгоритму ELECTRO
- Була обрана мова програмування, для реалізації ПП
- Була побудована діаграма послідовностей, даграма використання та інші, що будуть використані при реалізації ПП

## РОЗДІЛ 5

### РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 5.1. Етапи реалізації

Після завершення проектування ми знаємо з яких функціональних частин в нас буде побудований ПП, які властивостями він повинен володіти та що повинен отримати на вході та повертати на виході. Тому розробляємо послідовність етапів розробки:

1. Створення головного вікна програми та меню
2. Реалізація методу ELECTRE
3. Створення графічного інтерфейсу та методів вводу даних
4. Реалізація методу роботи з файлом
5. Реалізація методу виводу результатів

#### 5.2. Використані типи змінних та конструкції

У ПП був використана клас `list<>`. Через те що створенні масиви у C# мають фіксовану довжину, то для збереження даних необхідно було використати список `list<>` у який можна необмежену кількість разів додавати та витягати елементи. Він має наступні методи:

- **void Add(T item):** додає новий елемент у список
- **void AddRange(ICollection collection):** додавання у список колекції чи масиву
- **int BinarySearch(T item):** додавання бінарний пошук елементу у списку(лише для відсортованих масивів)
- **int IndexOf(T item):** повертає індекс елементу у списку(лише перше входження)
- **void Insert(int index, T item):** додає елемент item у список на певну позицію index
- **bool Remove(T item):** видаляє елемент item з списку

- **void RemoveAt(int index):** видаляє елемент за вказаним індексом index
- **void Sort():** сортує списку

Також була використана конструкція try...catch...finally – дуже ефективний спосіб пошуку несправностей у коді. Неважливо яку дію має виконувати програма ви можете помістити її у блок «try», якщо виконувана дія не викликала ніяких помилок то вона працює як треба і наступний блок не виконується. Якщо у результаті виконання виникла помилка, то програма не виконує усі дії у цьому блоці та переходить до блоку «catch» та виконує його, цей блок виловлює несправності у попередньому блоці і виконує себе при їх наявності. Наступний блок виконується завжди «finally».

```
try
{ //якась код
}
catch
{ //код який виконується
  //при помилці в try
}
finally
{ //код що виконується завжди
}
```

При розробці ПП цей блок був використаний для запобігання звернення для не існуючого елемента list<>.

```
try
{
dataGridView2.Rows[i].Cells[0].Value = name_alt[i];
}
catch
{
dataGridView2.Rows.Add(name_alt[i]);
}
```

### 5.3. Процес розрахунку індексів в програмі

Після того, як програма отримує базу даних альтернатив, вона обчислює значення двох таблиць розміром  $N^2$  для визначення індексів згоди та незгоди. Після цього вона зводить їх до  $N$  мірного масиву, який передається наступному блоку програми для виводу альтернатив. Через особливість статично типізованих мов програмування та необхідність зменшення кількості ітерацій розрахунків, таблиці в індексів знаходяться лише численні значення, а назви альтернатив та інша не чисельна інформація зберігається у окремих масивах. Нижче наведена таблиця індексів згоди яку обчислює програма.

Таблиця 5.1

**Таблиця індексів згоди що рахується у ПП**

	Alter_ _1	Alter_ _2	Alter_ _3	Alter_ _4	Alter_ _5	Alter_ _6	Alter_ _7	Alter_ _8	Alter_ _9	Alter_ _10
Alter_ _1	1	0,8	0,6	1	0,4	1	1	0,6	1	0,9
Alter_ _2	0,2	1	0,5	0,6	0,3	0,7	0,6	0,2	0,9	0,6
Alter_ _3	0,4	0,8	1	0,5	0,5	0,5	0,8	0,3	1	0,4
Alter_ _4	0	0,4	0,5	1	0,1	0,4	0,4	0,2	0,6	0
Alter_ _5	0,9	0,9	0,5	0,9	1	1	0,9	0,5	0,9	0,9
Alter_ _6	0	0,3	0,5	0,9	0,1	1	0,3	0,2	0,5	0,6
Alter_ _7	0,2	0,8	0,5	0,6	0,1	0,7	1	0,2	0,9	0,6
Alter_ _8	0,7	0,8	0,7	0,8	0,8	0,8	0,8	1	1	0,7
Alter_ _9	0	0,5	0	0,5	0,1	0,5	0,1	0,2	1	0,4
Alter_ _10	0,1	0,4	0,6	1	0,1	1	0,4	0,3	0,6	1



## 5.4. Аналіз даних

Для тестування програми були згенеровані випадкові дані та незалежно обчислені паралельно з програмою.

Таблиця 5.2

**Таблиця альтернатив з значенням ваг критеріїв**

Назва альтернатив	Вага критеріїв			
	1	2	3	4
Alter_1	7	7	10	6
Alter_2	2	10	5	3
Alter_3	6	1	5	8
Alter_4	3	3	6	0
Alter_5	0	10	10	7
Alter_6	0	6	6	1
Alter_7	2	7	5	5
Alter_8	5	0	10	9
Alter_9	3	0	3	3

У першій колонці - назва альтернатив, наступні чотири колонки починаються з ваги критерію і продовжуюся їм значення для кожної з альтернатив.

Таблиця 5.3

**Множина Парета на таблиці альтернатив 5.2**

Назва альтернатив	Вага критеріїв			
	1	2	3	4
Alter_1	7	7	10	6
Alter_2	2	10	5	3
Alter_3	6	1	5	8
Alter_5	0	10	10	7
Alter_7	2	7	5	5
Alter_8	5	0	10	9
Alter_10	8	6	9	1

На наданих даних програма зробила вірні розрахунки нових критеріїв, але глядячи на дані очевидно, що альтернативи номер 4,6,9 є аутсайдерами що програють по усім критеріям, отже їх можна видалити и нова таблиці матиме такий вигляд таблиці 5.3.

## 5.5. Windows Forms

При реалізації програмного продукту було використано API - Windows Forms. Windows Forms – це інтерфейс програмування додатку, відповідальний за графічний інтерфейс користувача та є частиною Microsoft .NET Framework. Спочатку, Windows Forms розглядалася лише як заміна, більш старої та складної бібліотеки Microsoft Foundation Classes, що була написана на C++. Але Windows Forms не передбачала парадигму Model-View-Controller. Для виправлення цієї ситуації та реалізації цієї можливості, існують сторонні бібліотеки. На відміну від пакетних програм, більша частина часу витрачається на очікування від користувача якихось дій, як ввід тексту у поле чи клацання мишкою.

Парадигма Model-View-Controller розумію під собою розділення даних додатку, користувацького інтерфейсу та управляючої логіки на три окремих компоненти. Таким чином їх можна буде модифікувати окремо один від одного.

## 5.6. DataGridView

У Microsoft Visual Studio елемент управління DataGridView, розроблений для використання в додатках, створених за шаблоном Windows Forms Application. Даний елемент управління дозволяє організовувати дані у

вигляді таблиці. Дані можуть отримуватись з бази даних, колекції, внутрішніх змінних – масивів чи інших об'єктів програми.

Для розробки ПП були була потрібна можливість зручно та зрозуміло показувати дані, також цей елемент управління досить, через те, що його можна використовувати як тимчасове місце збереження даних. Це дозволяє зменшити кількість змінних у коді ПП.

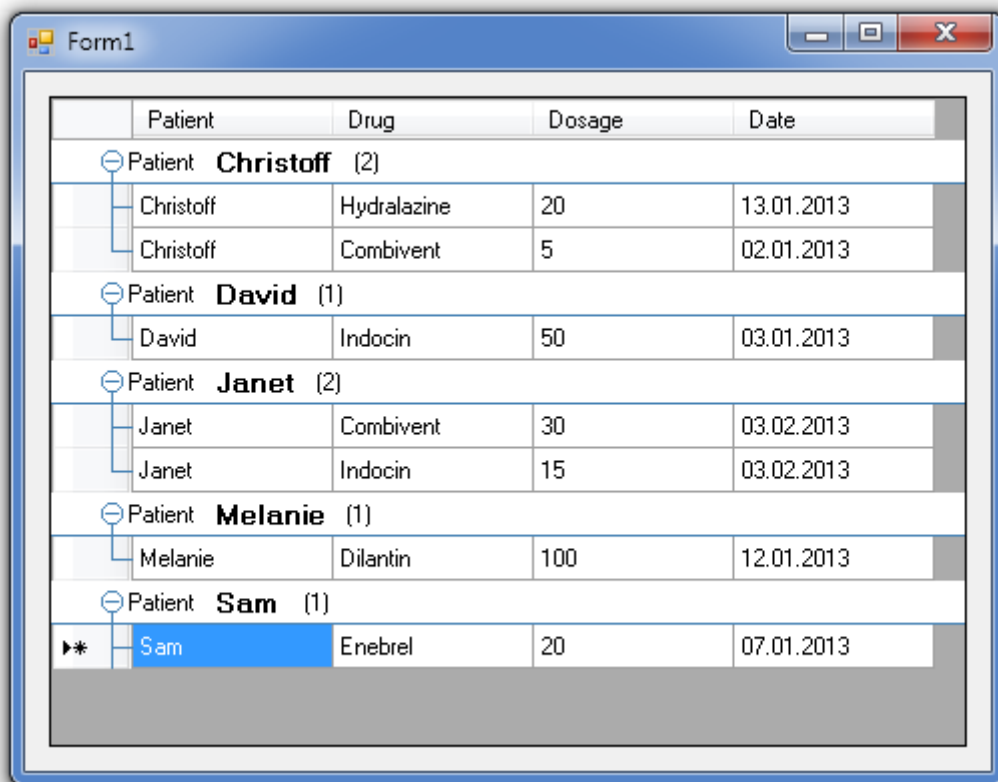


Рисунок 5.1. Приклад зовнішнього виду DataGridView

Головною перевагою DataGridView дає змогу отримувати з бази даних внутрішні структури даних, також процес переносу даних більш зручний для програміста та користувача. Також у комірках можуть бути представлені інші типи окрім строкових та численних перемінних, а саме:

- dataGridViewButtonColumn. Комірки представлені у вигляді кнопок типу Button;

- `dataGridViewCheckBoxColumn`. Комірки представлені елементами управління типу `CheckBox`, які дозволяють вибирати декілька варіантів (опцій) з набору запропонованих;
- `dataGridViewComboBoxColumn`. Комірки представлені елементами управління типу `ComboBox`, що призначені для вибору одного з декількох варіантів;
- `dataGridViewImageColumn`. Комірки таблиці є зображеннями типу `Image`;
- `dataGridViewLinkColumn`. Комірки таблиці представлені посиланнями;
- `dataGridViewTextBoxColumn`. Цей варіант пропонується за замовчуванням при додаванні (створенні) нової колонки. У цьому випадку комірки таблиці представлені у вигляді полів вводу. Це дозволяє вводити дані в таблицю як в матрицю.

За допомоги цього елемента у даному ПП був реалізований вивід даних на екран користувача, у вигляді таблиці та зміна ваг критеріїв та їх значень.

## 5.7. TrackBar

Один з важливих елементів ПП є `TrackBar`, що дозволяє за допомоги зміни положення повзунка вводити числові значення.

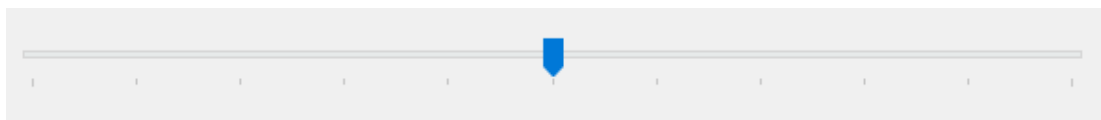


Рисунок 5.2. Зображення `TrackBar`

Він володіє наступними властивостями:

- `Orientation`: задає орієнтацію повзунка - расположение по горизонтали или по вертикали
- `TickStyle`: задає розміщення ділень на повзунку
- `TickFrequency`: задає частоту ділень на повзунку

- Minimum: мінімальне можливе значення на повзунку (за замовчуванням 0)
- Maximum: максимальне можливе значення на повзунку (за замовчуванням 10)
- Value: поточне значення повзунка. Повинно знаходитись між Minimum и Maximum

Даний елемент виконує роль важіля для зміни значень індексів згоди та незгоди, таким чином дозволяє користувачу порівняти альтернативи при різних їх значеннях.

## 5.8. Основні елементи ПП

Користувач запустивши програма бачить початкове вікно ПП. Воно представляє собою велике вікно з горизонтальним меню з трьома пунктами та великим полем для майбутньої таблиці. Тепер користувач відкриває меню пункт меню «Файл», та обирає підпункт «Відкрити файл».



Рисунок 5.3. Зображення стартового вікна ПП

Змін.	Лист	№ докум.	Підпис	Дата

**БС51.22.1300.1404с.ПЗ**

Лист

53

Відкривається провідник, де користувач може обрати файл з базою даних, що наявний на його комп'ютері, форматі \*.xlsx. Програма приймає на вхід базу даних з файлу, з найменуванням, значенням та вагою критеріїв альтернатив.

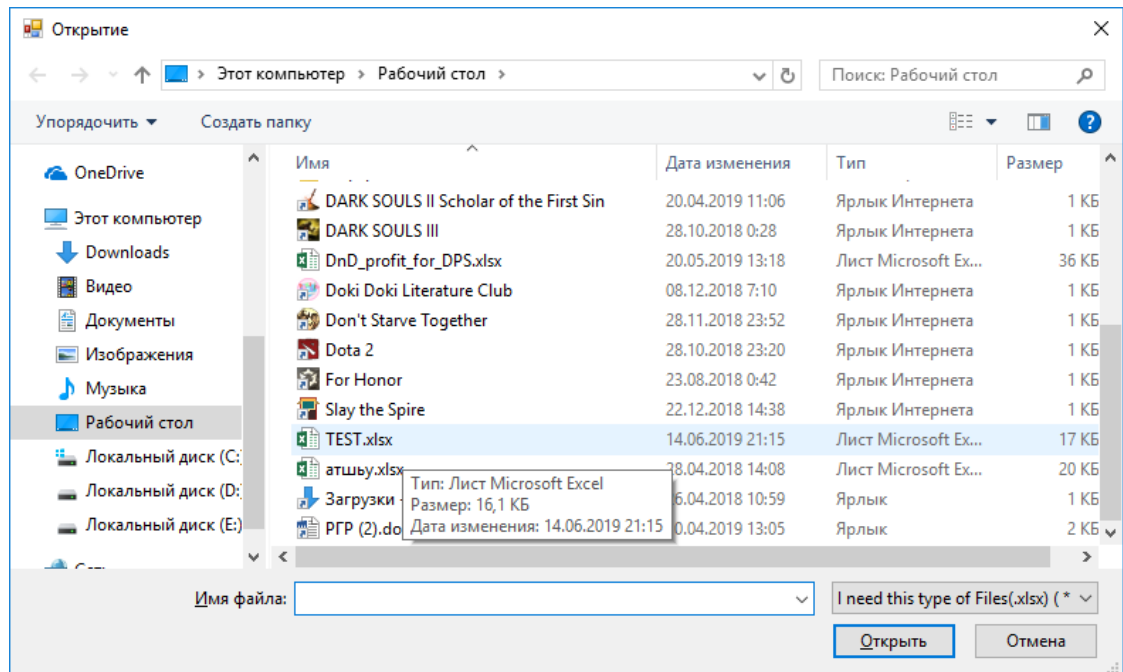


Рисунок 5.4. Вікно обору файлу з даними

Потім користувачу надається можливість корегувати дані, у попередньо зарезервованому просторі, та проаналізувати їх методом ELECTRO і, при необхідності зберегти результат роботи та правки над даними.

Після внесення право у данні, та обмеження критеріїв, користувач відкриває меню «Опції» та обирає пункт підменю «Алгоритм ELECTRE» - основну функцію програми.

	Назва альтернатив	ціна	розмір манжети	акумулятор	зручність
▶	#	2	3	1	4
	LONGEVITA BP-...	813	22	2	3
	DR. FREI M-110W	770	22	3	3
	DR. FREI M-150S	790	32	2	2
	MicroLife BP N1 B...	952	32	3	2
	LONGEVITA LS-5	338	36	4	1
*					

Рисунок 5.5. Звантажена таблиця даних у стартовому вікні ПП

Пункт підменю «Алгоритм ELECTRE», запускає на виконання алгоритм підрахунку індексів методу ELECTRE. Відкривається нове вікно на якому зображено два підписані повзунки таблиця. Користувач може рухати повзунки, таким чином змінюючи допустиме значень індексів.

	Альтернатива	Відношення	Точність	Відношення
▶	Alter_1	7	7	10
	Alter_2	2	10	5
	Alter_3	6	1	5
	Alter_4	3	3	6
	Alter_5	0	10	10

Рисунок 5.6. – підменю пункта «Опції», головного меню

При зміні індексів згоди та незгоди альтернативи порівнюються і змінюються значення таблиці бінарних відносин та вага альтернатив.

The screenshot shows a software window titled "Form2" with two sliders at the top. The first slider, labeled "Індекс згоди", has a blue arrow pointing to the right. The second slider, labeled "Індекс незгоди", has a blue arrow pointing to the left. Below the sliders is a table titled "Таблиця ваг альтернатив".

	Назва альтернативи	A1	A2	A3	A4	A5	Вага альтернативи
	LONGEVIT...	-	0	0	0,5	0,5	1
	DR. FREI ...	1	-	0,5	0,5	0,5	2,5
	DR. FREI ...	1	0,5	-		0,5	3
	Microlife B...	0,5	0,5	0	-	0,5	1,5
	LONGEVIT...	0,5	0,5	0,5	0,5	-	2
*							

Рисунок 5.7. Вікно з зваженими альтернативами

Змінюючи значення індексів згоди та незгоди ми бачимо як змінюється вага різних альтернатив.



## Висновки до розділу 5

В даному розділі описана реалізація ПП, були наведені та описані властивості елемент, що були використані під час реалізації ПП. Програма ула протестована на реальних даних, та виводить вірні результати. Реалізовано:

- Завантаження бази даних та її збереження у файл
- Ввід параметрів для завантажених даних, та можливість їх зміни
- Розрахунок ваг альтернатив методом ELECTRE

					<b>БС51.22.1300.1404с.ПЗ</b>	Лист
Змін.	Лист	№ докум.	Підпис	Дата		57

## РОЗДІЛ 6

### ОХОРОНА ПРАЦІ

#### Вступ

Метою дипломного проекту є розробка додатку для підтримки прийняття рішення про закупівлю медичного обладнання за сукупністю критеріїв

У цьому розділі охарактеризовано кабінет відділу постач, в якому буде використовуватися даний додаток, з метою визначення всіх можливих небезпек фізичного, хімічного, електричного та пожежного походження, а також описати всі можливі заходи щодо усунення виявлених небезпек або зменшення їх впливу на працівників.

#### 6.1. Загальна характеристика приміщення

Характеристика об'єктів приміщення вказані в табл. 6.1-6.2.

Таблиця 6.1

##### Параметри приміщення

№	Найменування	Основні характеристики	Кількість	Позиція на рисунок
1	Параметри приміщення	3000 (мм) x 5000 (мм) x 3000 (мм); S=15 м <sup>2</sup> ; V=45 м <sup>3</sup>		-
2	Кількість працюючих	Керівник, помічник	2	-
3	Природне освітлення	Вікно 1900x1500 мм	1	-
4	Штучне освітлення	Стельова лампа днівного світлу	4	1

## Обладнання і оснащення

№	Назва	Основні характеристики	К-сть	Поз. на рис.
5	Крісло AMF Ровер Tilt Неаполь N-20	Екологічна шкіра: 25% бавовна (CO), 75% поліестер (PES); 540х480х620	2	2
8	Тумбочка ОН-38-20	Дерев'яний(ЛДСП) 400х350х650	1	3
9	Стіл СКТ-4	Дерев'яний(ЛДСП) 1400х900х750	3	4
10	Принтер Canon PIXMA E204 (2320C009AA)	Струменевий принтер, Споживана потужність при друку 10 Вт	1	5
11	ПК ARTLINE Home H38 v04 (H38v04)	405 x 180 x 410 мм, Двоядерний Intel Pentium G4560 (3.5 ГГц), HDD 1 ТБ, Оперативна пам'ять 8 ГБ, Чипсет Intel B250	1	6
12	Дошка для заміток	Дошка магнітна під маркер, сухостираєма 1000 x 3000 мм	1	7
13	Датчик пожежний димовий ДИП-34АВТ	Чутливість 0,05-0,2 дБ/м 102 x 35мм	2	8
14	Вогнегасник ОП-10	Вогнегасник порошковий, 170 x 650 мм, 14кг	1	9
5	Крісло AMF Ровер Tilt Неаполь N-20	Екологічна шкіра: 25% бавовна (CO), 75% поліестер (PES); 540х480х620	2	2
8	Тумбочка ОН-38-20	Дерев'яний(ЛДСП) 400х350х650	1	3
9	Стіл СКТ-4	Дерев'яний(ЛДСП) 1400х900х750	3	4
10	Принтер Canon PIXMA E204 (2320C009AA)	Струменевий принтер, Споживана потужність при друку 10 Вт	1	5

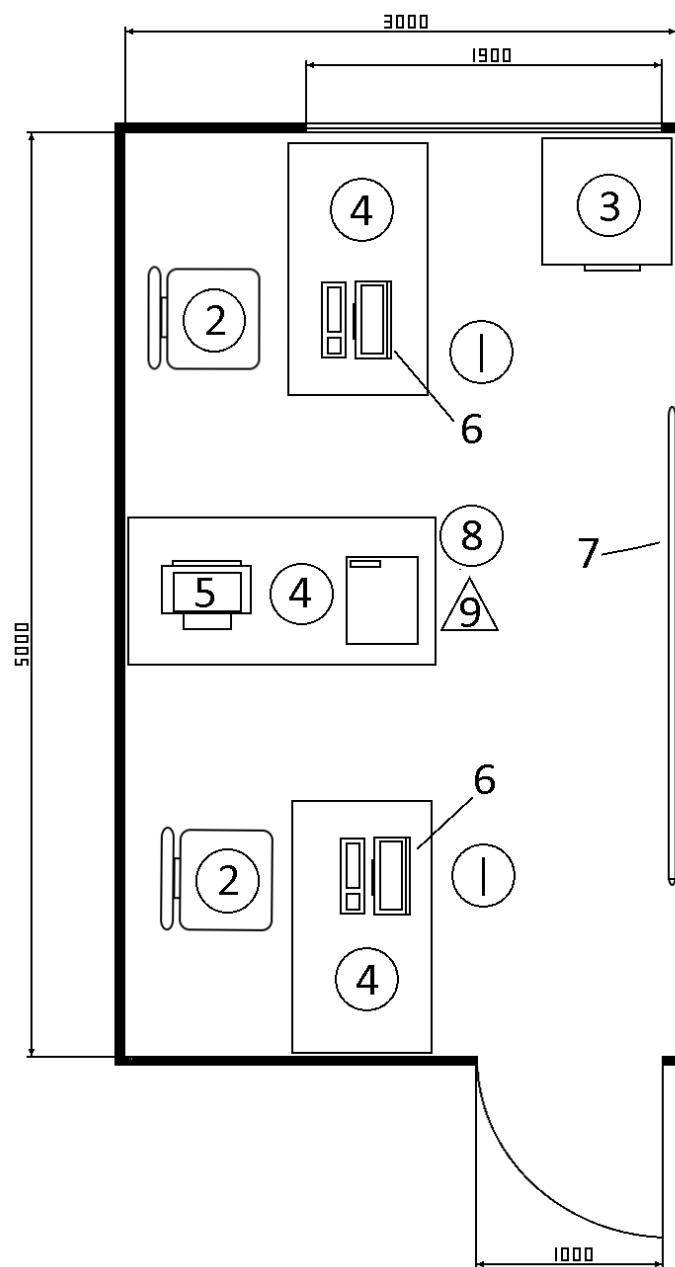


Рисунок 6.1. План схема кабінету відділу постач

Таблиця 6.3

### Порівняння фактичних та нормативних характеристик

№	Параметр приміщення	Реальне значення	Нормативні значення
1	Площа на 1 працюючого	7,5 м <sup>2</sup>	4,5 м <sup>2</sup>
2.	Об'єм на 1 працюючого	22,5 м <sup>3</sup>	15 м <sup>3</sup>
3	Мінімальна ширина проходу	1,5	1,5 м

Заходів нормалізації не потрібно. За НАПБ Б.03.002-2007 обсяг приміщення на одну людину більший нормативного значення

## 6.2. Електробезпека

Приміщення оснащене електричними приладами. Використовується мережа однофазного струму 220В. За класифікацією кабінет відноситься до класу приміщень без підвищеної небезпечності ураження персоналу електричним струмом. Потенційно небезпечне оснащення в приміщення вказане в табл. 6.4.

Таблиця 6.4

### Характеристика електробезпеки

№	Найменування	Джерело небезпеки	Причини небезпеки	Наслідки небезпеки
1	ПК	Блок живлення	Несправності блоку живлення	Ураження струмом
2	Монітор	Кабель живлення	Пробій в кабелі живлення	
3	Принтер	Кабель живлення	Пробій в кабелі живлення	

Джерела ураження струмом вказані в табл. 6.5.

Таблиця 6.5

### Реальні та нормативні фактори небезпеки

№	Фактор небезпеки	Реальне значення	Нормативні значення
1	Максимальний струм	>1 А	0,25 А
		220 В	45 В

Для зниження ймовірності настання небезпечної ситуації, необхідно дотримуватись заходів безпеки, які наведені в табл. 6.6.

**Засоби захисту від електротравм**

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи Організаційні заходи	Ящик ЯРП-100А 74 У1 IP54	Попереджує небезпечні ситуації під час перепаду напруги
2.	Організаційні заходи	Проведення інструктажів з правил роботи з ПК	Навчання з питань електричної безпеки при експлуатації обладнання
3.	Експлуатаційні	Проведення планових ремонтів електроприладів	Зменшення імовірності ураження електричним струмом

Виявлена наявність електробезпеки, яка може проявлятися у вигляді замикання електромережі та надання травм організму при контакті з джерелом небезпеки.

**6.3. Пожежна безпека**

У приміщенні наявні займисті речовини: папір, дерево або ДСП, пластик та електрична техніка, тому дане приміщення відноситься до категорії В, пожежонебезпечної зони класу П-Па. Джерелом займання може бути коротке замикання та несправність електромережі.

Таблиця 6.7

**Джерела пожежонебезпеки**

№	Група номенклатурних заходів з ОП	Джерело небезпеки	Причини небезпек	Наслідки небезпеки
1	ПК	Блок живлення	Коротке замикання	Опіки, отруєння димом
2	Лампа	Проводка	Пошкодження обмотки	Опіки, отруєння димом
3	Принтер	Папір, пил, елементи друкувальної частини	Коротке замикання	Отруєння димом
4	Монітор	Мікросхеми	Коротке замикання	Опіки, отруєння димом
5	Столи	Зовнішнє	Довготривала дія високих температур	Опіки, отруєння димом

Характеристика пожежонебезпеки зазначена в табл. 6.8.

Таблиця 6.8

**Фактори небезпеки приміщення**

№	Фактор небезпеки	Реальне значення	Нормативні значення
1	Відкрите полум'я	-	0
2	Дим	-	0

В приміщенні необхідні наступні засоби безпеки (табл. 6.9).

**Засоби та заходи захисту від пожежі**

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	Вогнегасник ОП-10	Гасіння первинних осередків займання
		Димових датчика ДИП-34АВТ	Індикація наявності диму
2.	Організаційні заходи	Інструктаж з пожежної безпеки	Навчання з питань безпеки при пожежі
		План евакуації при пожежі	Для запобігання людських жертв при пожежі
3.	Експлуатаційні	Проведення планових ремонтів електроприладів	Захист від пожеж, що викликані несправностями техніки

Для забезпечення пожежної безпеки у приміщенні присутній порошковий вогнегасник та димовий пожежний сповіщувач.

**Висновки до розділу 6**

В цьому розділі дипломної роботи було розглянуто норми та заходи щодо охорони праці та техніки безпеки в кабінеті відділу постач. Були проаналізовані можливі потенційні небезпеки та виробничі фактори на робочому місці під час експлуатації технічного обладнання.

Дане приміщення знаходиться в допустимих експлуатаційних значеннях по електробезпеці, пожежі при дотриманні необхідної техніки безпеки. Також приміщення має належні параметри площі та об'єму.



## РОЗДІЛ 7

### ЕКОНОМІЧНА ЧАСТИНА

#### 1.1. Економічний аналіз варіантів розробки програмного продукту

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Загальна трудомісткість обчислюється як:  $T_o = T_p * K_p * K_{ск} * K_m * K_{ст} * K_{ст.м}$ ,

де  $T_p$  – трудомісткість розробки програмного продукту;  $K_p$  – поправочний коефіцієнт;  $K_{ск}$  – коефіцієнт на складність вхідної інформації;  $K_m$  – коефіцієнт рівня мови програмування;  $K_{ст}$  – коефіцієнт використання стандартних модулів і прикладних програм;  $K_{ст.м}$  – коефіцієнт стандартного математичного забезпечення.

Розробка програмного продукту включає в себе 3 завдання.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеня новизни Б та групи складності алгоритму 1, трудомісткість  $T_p = 90$  людино-днів. Поправочний коефіцієнт, який враховує вид використаної інформації для першого завдання (нормативно-довідкова інформація):  $K_{пк} = 1,35$ . Поправковий коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для завдань,  $K_{ск} = 1$ .

Оскільки під час розробки першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта  $K_{ст} = 0,7$ . Коефіцієнти  $K_m$  і  $K_{ст.п}$ , які враховують відповідно програмування на мові низького рівня та розробку стандартного програмного забезпечення, для всіх завдань становлять 1:  $K_m = K_{ст.п} = 1$ .

Отже, загальна трудомісткість програмування першого завдання варіанту дорівнює:

$$T_o = T_p \cdot K_p \cdot K_{ск} \cdot K_{ст} = 90 \cdot 1,35 \cdot 1 \cdot 0,7 = 85,05 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для інших завдань.

Для другого завдання (використовується алгоритм першої групи складності, ступінь новизни Б):

$$T_p = 90 \text{ людино-днів; } K_p = 1,08; K_{ст} = 0,8;$$

$$T_o = 90 \cdot 1,08 \cdot 0,8 = 77,76.$$

Для третього завдання варіант А (використовується алгоритм першої групи складності, ступінь новизни В):

$$T_p = 60 \text{ людино-днів; } K_p = 1,02; K_{ст} = 0,8;$$

$$T_o = 60 \cdot 1,02 \cdot 0,8 = 48,96.$$

Для третього завдання Варіант Б (використовується алгоритм другої групи складності, ступінь новизни В):

$$T_p = 19 \text{ людино-днів; } K_p = 0,81; K_{ст} = 0,8;$$

$$T_o = 19 \cdot 0,81 \cdot 0,8 = 12,31.$$

Складемо трудомісткість відповідних завдань, щоб отримати їх трудомісткість:  $T_I = (85,05 + 77,76 + 48,96) \cdot 8 = 1694,16 \text{ людино-годин}$

$$T_{II} = (85,05 + 77,76 + 12,31) \cdot 8 = 1400,96 \text{ людино-годин}$$

Очевидно, що вищу трудомісткість має варіант І.

## 1.2. Розрахунок заробітної плати розробника

В розробці беруть участь два програмісти з окладом 20000 грн. Денну заробітню плату визначимо, виходячи із місячних окладів програмістів, враховуючи тривалість умовного місяця 21 день при 5-ти робочих днів на тиждень. Визначимо зарплату за годину за формулою:  $C_{г} = \frac{M}{T_m \cdot t} \text{ грн.,}$

де М – місячний оклад працівників;  $T_m$  – кількість робочих днів тиждень; t – кількість робочих годин в день.

$$C_{г} = \frac{20000 + 20000}{21 \cdot 8 \cdot 2} = 119,05 \text{ грн.}$$

					<b>БС51.22.1300.1404с.ПЗ</b>	Лист
Змін.	Лист	№ докум.	Підпис	Дата		66

Тоді, розрахуємо заробітну плату за формулою

$S_{зп} = S_{ч} \cdot T_i \cdot K_D$ , де  $S_{ч}$  – величина погодинної оплати праці програміста;  $T_i$  – трудомісткість відповідного завдання;  $K_D$  – норматив, який враховує додаткову заробітну плату.

Тоді зарплата розробників за варіантами відповідно до формули:

I.  $S_{зп} = 119,05 \cdot 1694,16 \cdot 1,2 = 242027,7$  грн.

II.  $S_{зп} = 119,05 \cdot 1400,96 \cdot 1,2 = 200141,15$  грн.

Відрахування на всі види соціального страхування за варіантами (22%):

I.  $S_{від} = 242027,7 \cdot 0,22 = 53246,09$  грн.

II.  $S_{від} = 200141,15 \cdot 0,22 = 44031,05$  грн.

Тепер визначимо витрати на оплату однієї машино-години. ( $S_M$ )

Так як одна ЕОМ обслуговує одного програміста з окладом 20000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$S_M = 12 \cdot M \cdot K_3 = 12 \cdot 20000 \cdot 0,2 = 48000$  грн.

З урахуванням додаткової заробітної плати:

$S_{зп} = S_M \cdot (1 + K_3) = 48000 \cdot (1 + 0,2) = 57600$  грн.

Відрахування на єдиний соціальний внесок:

$S_{від} = S_{зп} \cdot 0,367 = 57600 \cdot 0,367 = 21139,2$  грн.

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 25000 грн.

$S_a = K_{тм} \cdot K_a \cdot Ц_{пр} = 1,15 \cdot 0,25 \cdot 25000 = 7187,5$  грн.,

де  $K_{тм}$  – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;  $K_a$  – річна норма амортизації;  $Ц_{пр}$  – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$S_r = K_{тм} \cdot Ц_{пр} \cdot K_r = 1,15 \cdot 25000 \cdot 0,05 = 1437,5$  грн.,

де  $K_r$  – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{эф}} = (D_k - D_v - D_c - D_p) \cdot t_z \cdot K_v = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4$$

годин,

де  $D_k$  – календарна кількість днів у році;  $D_v, D_c$  – відповідно кількість вихідних та святкових днів;  $D_p$  – кількість днів планових ремонтів

устаткування;  $t$  – кількість робочих годин в день;  $K_v$  – коефіцієнт використання

приладу у часі протягом зміни.

Витрати на оплату електроенергії для другого класу напруги розраховуємо за формулою:

$$C_{\text{ел}} = T_{\text{эф}} \cdot N_c \cdot K_z \cdot C_{\text{ен}} = 1706,4 \cdot 0,156 \cdot 2 \cdot 278,39 = 1482,14 \text{ грн.}$$

де  $N_c$  – середньо-споживча потужність приладу;  $K_z$  – коефіцієнтом зайнятості приладу;  $C_{\text{ен}}$  – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_n = C_{\text{пр}} \cdot 0.67 = 25000 \cdot 0,67 = 16750 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{екз}} = C_{\text{зп}} + C_{\text{від}} + C_a + C_p + C_{\text{ел}} + C_n = 57600 + 21139,2 + 7187,5 + 1437,5 + 1482,14 + 16750 = 105596,34 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{м-г}} = C_{\text{екз}} / T_{\text{эф}} = 97129,14 / 1706,4 = 56,92 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу,

в залежності від обраного варіанта реалізації, складає:

$$C_{\text{м.ч}} = C_{\text{м-г}} \cdot T$$

$$\text{I. } C_{\text{м.ч}} = 56,92 \cdot 1694,16 = 96431,59 \text{ грн.}$$

$$\text{II. } C_{\text{м.ч}} = 56,92 \cdot 1400,96 = 79742,64 \text{ грн.}$$

Накладні витрати становлять 67 % від заробітної плати:

$$C_H = C_{3П} \cdot 0,67$$

$$I. C_{\text{накл}} = 242027,7 \cdot 0,67 = 162158,56 \text{ грн.}$$

$$II. C_{\text{накл}} = 200141,15 \cdot 0,67 = 134094,57 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{ПП}} = C_{3П} + C_{\text{від}} + C_M + C_H$$

$$I. C_{\text{ПП}} = 242027,7 + 53246,09 + 96431,59 + 162158,56 = 553863,94 \text{ грн.};$$

$$II. C_{\text{ПП}} = 200141,15 + 44031,05 + 79742,64 + 134094,5 = 458009,34 \text{ грн.}$$

### Висновки до розділу 7

В даному розділі було розраховано загальну трудомісткість програмування для трьох різних варіантів завдання, заробітну плату робітника, а також вартість розробки програмного продукту.

## ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання дипломної роботи було розроблену ПП, що дозволяє обирати найкращу альтернативу серед існуючих, використовуючи для цього алгоритм багатокритеріального вибору ELECTRE. Програм було реалізовано у середі розробки Microsoft Visual Studio, на об'єктно-орієнтованій мові програмування C#, з використанням базових бібліотек. У програму були додані можливості різного налаштування та обробки даних включно зі зміною окремих елементів.

Були вирішені поставлені на початку завдання:

1. Проведений аналіз вітчизняних та закордонних літературних джерел з поставленої проблематики. Були підкреслені певні та реалізовані певні особливості обрахунку різних методів, та сформульовані визначення та принципи у ТПП. Визначено, що метод R

2. Обґрунтувати використання методу ELECTRE, як головного методу пошуку кращої альтернативи, шляхом порівняння його з іншими методами багатокритеріального обору. Визначено, що метод ELECTRE є оптимальним для його програмної реалізації, через розрахунок ваги альтернатив, шляхом попарного порівняння.

3. Розроблений ПП для обору найкращої альтернативи на основі методу ELECTRE з зручним інтерфейсом. Інтерфейс ПП дозволяє у будь-який момент часу роботи продивитись дані, що будуть оцінюватись, змінювати її, вводити обмеження та повністю використовувати властивості методу ELECTRE. Програма може самодостатньо працювати на операційній системі Windows.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Файнзільберг Л.С., Жуковська О.А., Якимчук В.С. Теорія прийняття рішень
2. Васильченко І.П. Вища математика: основні означення, приклади і задачі
3. Дмитриенко В. Д. Введение в теорию и методы принятия решений
4. Кушлик-Дивульська О.І. Теорія ймовірностей та математична статистика
5. Де Гроот М. Оптимальные статистические решения (пер. с англ.)
6. Орлов А. И. Теория принятия решений: учебник.
7. Орлов А. И. Принятие решений. Теория и методы разработки управленческих решений. Учебное пособие.
8. Sven Ove Hansson. Decision Theory: A Brief Introduction,
9. Ногин В. Д. Принятие решений в многокритериальной среде
10. M. Ehrgott and X. Gandibleux. Approximative Solution Methods for Multiobjective Combinatorial Optimization
11. Ivanenko V.I. Decision Systems and Nonstochastic Randomness
12. Файнзильберг Л.С. Математические методы оценки полезности диагностических признаков
13. Подиновский В. В. Парето – оптимальные решения многокритериальных задач
14. Литвак Б.Г. Экспертная информация: Методы получения и анализа
15. Блауг М. Экономическая теория благосостояния Парето
16. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач

17. Кини Р.Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения
18. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход
19. Matthias Ehrgott. Multicriteria Optimization
20. Ларичев О.И. Теория и методы принятия решений
21. Литвак Б. Г. Разработка управленческого решения — М.: Издательство «Дело», 2004 г. — 392 с.
22. Paul Goodwin and George Wright, Decision Analysis for Management Judgment, 3rd edition. Chichester: Wiley, 2004
23. Robert Clemen. Making Hard Decisions: An Introduction to Decision Analysis, 2nd edition. Belmont CA: Duxbury Press, 1996.
24. Morris De Groot Optimal Statistical Decisions. Wiley Classics Library. 2004.
25. Howard Raiffa Decision Analysis: Introductory Readings on Choices Under Uncertainty. McGraw Hill. 1997.
26. Glenn Shafer and Judea Pearl, editors. Readings in uncertain reasoning. Morgan Kaufmann, San Mateo, CA, 1990.
27. J.Q. Smith Decision Analysis: A Bayesian Approach. Chapman and Hall.
28. Akerlof, George A. and Janet L. YELLEN, Rational Models of Irrational Behavior
29. James O. Berger Statistical Decision Theory and Bayesian Analysis. Second Edition. 1980. Springer Series in Statistics
30. Arthur, W. Brian, Designing Economic Agents that Act like Human Agents: A Behavioral Approach to Bounded Rationality
31. Mas-Colell, Andreu; Whinston, Michael; & Green, Jerry (1995). Microeconomic Theory. Oxford:Oxford University Press.



32. . А. Козицький, С. П. Лавренюк, М. О. Оліскевич (2004). Основи математичної економіки. Теорія споживання.
33. О. І. Пономаренко, М. О. Перестюк, В. М. Бурим (2004). Сучасний економічний аналіз. Мікроекономіка.
34. Вебер Г., Вельштейн И. (ред.) Энциклопедия элементарной математики. Том 1. Элементарная алгебра и анализ. Одесса: Матезис, 1906
35. Довідковий математичний словник: Для студ. вузів екон. спрямування / Г. Я. Дутка; Нац. банк України. — Л., 1998. — 95 с.
36. Довідник з елементарної математики. Арифметика, алгебра / К. І. Швецов, Г. П. Бевз. — К.: Наукова думка, 1967. — 408 с.
37. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. — М. : Наука, 1974. — 832 с.
38. Короткий тлумачний математичний словник / Бугай А. С. — К.: Радянська школа, 1964. — 428 с.
39. Кісілевич О. В., Пенцак О. С., Барбуляк Л. В. Математика. — Львів : Новий Світ-2000, 2006. — 320 с.
40. Курант Р., Роббинс Г. Что такое математика? — 3-е изд., испр. и доп.—М.: МЦНМО, 2001. — 568 с.
41. Кини Р.Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения. — М: Радио и связь, 1981. — 560 с.
42. Matthias Ehrgott. Multicriteria Optimization. — Springer, 2005.
43. Davis G. Management Information Systems: Conceptual Foundations, Structure, and Development. — New York: McGraw-Hill, 1974.
44. Marakas G. M. Decision support systems in the twenty-first century. Upper Saddle River, N.J.: Prentice Hall, 1999.
45. Power D. J. «What is a DSS?» // The On-Line Executive Journal for Data-Intensive Decision Support, 1997. — v. 1. — N3

